UPAL: Unbiased Pool Based Active Learning

Ravi Ganti **Alexander Gray**
College of Computing, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

In this paper we address the problem of pool based active learning, and provide an algorithm, called UPAL, that works by minimizing the unbiased estimator of the risk of a hypothesis in a given hypothesis space. For the space of linear classifiers and the squared loss we show that UPAL is equivalent to an exponentially weighted average forecaster. Exploiting some recent results regarding the spectra of random matrices allows us to analyze UPAL with squared losses for the noiseless setting. Empirical comparison with an active learner implementation in Vowpal Wabbit, and a previously proposed pool based active learner implementation show good empirical performance and better scalability.

1 Introduction

In the problem of binary classification one has a distribution $\mathcal{D}$ on the domain $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^d \times \{-1, +1\}$, and access to a sampling oracle, which provides us i.i.d. labeled samples $\mathcal{S} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. The task is to learn a classifier h, which predicts well on unseen points. For certain problems the cost of obtaining labeled samples can be quite expensive. For instance consider the task of speech recognition. Labeling of speech utterances needs trained linguists, and can be a fairly tedious task. Similarly in information extraction, and in natural language processing one needs expert annotators to obtain labeled data, and gathering huge amounts of labeled data is not only tedious for the experts but also expensive. In such cases it is of interest to design learning algorithms, which need only a few labeled examples for training, and also guarantee good performance on unseen data.

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

Suppose we are given a labeling oracle $\mathcal{O}$, which when queried with an unlabeled point x returns the label y of x. Active learning algorithms query this oracle as few times as possible and learn a provably good hypothesis from these labeled samples. Broadly speaking active learning (AL) algorithms can be classified into three kinds, namely membership query (MQ) based algorithms, stream based algorithms and pool based algorithms. All these three kinds of AL algorithms query the oracle $\mathcal{O}$ for the label of the point, but differ from each other in the nature of the queries. In MQ based algorithms the active learner can query for the label of a point in the input space $\mathcal{X}$, but it might not necessarily be from the support of the marginal distribution $\mathcal{D}_{\mathcal{X}}$. With human annotators MQ algorithms might work poorly as was demonstrated by Lang and Baum in the case of handwritten digit recognition (1992), where the annotators were faced with the awkward situation of labeling semantically meaningless images. Stream based AL algorithms (Cohn et al., 1994; Chu et al., 2011) sample a point x from the marginal distribution $\mathcal{D}_{\mathcal{X}}$, and decide on-the-fly whether to query $\mathcal{O}$ for the label of x? Stream based AL algorithms tend to be computationally efficient, and most appropriate when the underlying distribution changes with time. Pool based AL algorithms assume that one has access to a large pool $\mathcal{P} = \{x_1, \ldots, x_n\}$ of unlabeled i.i.d. examples sampled from $\mathcal{D}_{\mathcal{X}}$, and given budget constraints B, the maximum number of points they are allowed to query, query the most informative set of points. Both pool based AL algorithms, and stream based AL algorithms overcome the problem of awkward queries, which MQ based algorithms face. However in our experiments we discovered that stream based AL algorithms tend to query more points than necessary, and have poorer learning rates when compared to pool based AL algorithms.

Contributions. In this paper we propose a pool based active learning algorithm called UPAL, which given a hypothesis space $\mathcal{H}$, and a margin based loss function $\phi(\cdot)$ minimizes a provably unbiased estimator of the risk $\mathbb{E}[\phi(yh(x))]$. While unbiased estimators of risk have been used in stream based AL algorithms, no

such estimators have been introduced for pool based AL algorithms. We do this by using the idea of importance weights introduced for AL in Beygelzimer et al. (Beygelzimer et al., 2009). Roughly speaking UPAL proceeds in rounds and in each round puts a probability distribution over the entire pool, and samples a point from the pool. It then queries for the label of the point. The probability distribution in each round is determined by the current active learner obtained by minimizing the importance weighted risk over $\mathcal{H}$. Specifically in this paper we shall be concerned with linear hypothesis spaces.

In theorem 2 (Section 2.1) we show that for the squared loss UPAL is equivalent to an exponentially weighted average (EWA) forecaster commonly used in the problem of learning with expert advice (Cesa-Bianchi and Lugosi, 2006). Precisely we show that if each hypothesis h in $\mathcal{H}$ is considered to be an expert and the importance weighted loss on the currently labeled part of the pool is used as an estimator of the risk of $h \in \mathcal{H}$, then the hypothesis learned by UPAL is the same as an EWA forecaster. Hence UPAL can be seen as pruning the hypothesis space, in a soft manner, by placing a probability distribution that is determined by the importance weighted loss of each classifier on the currently labeled part of the pool.

In section 3 we prove consistency of UPAL with the squared loss for the noiseless setting, when the true underlying model is a linear model. Our proof employs some elegant results from random matrix theory regarding eigenvalues of sums of random matrices (Hsu et al., 2011a,b; Tropp, 2010). While it should be possible to improve the constants and exponents in theorem 3, our results qualitatively provide us the insight that in the noiseless setting the label complexity with the squared loss will depend on the condition number, and the minimum eigenvalue of the covariance matrix Σ. This kind of insight to our knowledge has not been provided before in the literature of active learning. In section 5 we provide a thorough empirical analysis of UPAL comparing it to the active learner implementation in Vowpal Wabbit (VW) (Langford et al., 2011), and a batch mode active learning algorithm, which we shall call as BMAL (Hoi et al., 2006). These experiments demonstrate the positive impact of importance weighting, and the better performance of UPAL over the VW implementation. We also empirically demonstrate the scalability of UPAL over BMAL on the MNIST dataset. When provided with a large budget UPAL is upto 7 times faster than BMAL.

2 Algorithm Design

A good active learning algorithm needs to take into account the fact that the points it has queried might not reflect the true underlying marginal distribution. This problem is similar to the problem of dataset shift (Quinonero et al., 2008) where the train and test distributions are potentially different, and the learner needs to take into account this bias during the learning process. One approach to this problem is to use importance weights, where during the training process instead of weighing all the points equally the algorithm weighs the points differently. UPAL proceeds in rounds, where in each round t, we put a probability distribution $\{p_i^t\}_{i=1}^n$ on the entire pool $\mathcal{P}$, and sample one point from this distribution. If the sampled point was queried in one of the previous rounds $1, \ldots, t-1$ then its queried label from the previous round is reused, else the oracle $\mathcal{O}$ is queried for the label of the point. Denote by $Q_i^t \in \{0, 1\}$ a random variable that takes the value 1 if the point x_i was queried for it's label in round t and 0 otherwise. In order to guarantee that our estimate of the error rate of a hypothesis $h \in \mathcal{H}$ is unbiased we use importance weighting, where a point $x_i \in \mathcal{P}$ in round t gets an importance weight of $\frac{Q_i^t}{p_i^t}$. By the definition of the random variable Q_i^t, we get $\mathbb{E}[Q_i^t | p_i^t] = 1$. We formally prove that importance weighted risk is an unbiased estimator of the true risk. Let $\mathcal{D}_n$ denote a product distribution on $(x_1, y_1), \ldots, (x_n, y_n)$. Also denote by $Q_{1:n}^{1:t}$ the collection of random variables $Q_1^1, \ldots, Q_n^1, \ldots, Q_n^t$. Let $\langle \cdot, \cdot \rangle$ denote the inner product. We have the following result.

Theorem 1. *Let* $\hat{L}_t(h) \stackrel{\text{def}}{=} \frac{1}{nt} \sum_{i=1}^n \sum_{\tau=1}^t \frac{Q_i^\tau}{p_i^\tau} \phi(y_i \langle h, x_i \rangle)$, *where* $p_i^\tau > 0$ *for all* $i = 1, \ldots, n, \tau = 1, \ldots, t$. *Then* $\mathbb{E}_{Q_1^1, \ldots, Q_n^t, \mathcal{D}_n} \hat{L}_t(h) = L(h)$.

Proof.

$$\begin{aligned}
\mathbb{E}_{Q_{1:n}^{1:t}, \mathcal{D}_n} \hat{L}_t(h) &= \mathbb{E}_{Q_{1:n}^{1:t}, \mathcal{D}_n} \frac{1}{nt} \sum_{i=1}^n \sum_{\tau=1}^t \frac{Q_i^\tau}{p_i^\tau} \phi(y_i \langle h, x_i \rangle) \\
&= \mathbb{E}_{Q_{1:n}^{1:t}, \mathcal{D}_n} \frac{1}{nt} \sum_{i=1}^n \sum_{\tau=1}^t \mathbb{E}_{Q_i^\tau | Q_{1:n}^{1:\tau-1}, \mathcal{D}_n} \\
&\qquad \frac{Q_i^\tau}{p_i^\tau} \phi(y_i \langle h, x_i \rangle) \\
&= \mathbb{E}_{\mathcal{D}_n} \frac{1}{nt} \sum_{i=1}^n \sum_{\tau=1}^t \phi(y_i \langle h, x_i \rangle) = L(w). \quad \square
\end{aligned}$$

The theorem guarantees that as long as the probability of querying any point in the pool in any round is non-zero, $\hat{L}_t(h)$, will be an unbiased estimator of

$L(h)$. How does one come up with a probability distribution on $\mathcal{P}$ in each round? To solve this problem we resort to probabilistic uncertainty sampling, where the point whose label is most uncertain as per the current hypothesis, $h_{A,t-1}$, gets a higher probability mass. The current hypothesis is simply the minimizer of the importance weighted risk in $\mathcal{H}$, i.e. $h_{A,t-1} = \arg\min_{h\in\mathcal{H}} \hat{L}_{t-1}(h)$. For any point $x_i \in \mathcal{P}$, to calculate the uncertainty of the label y_i of x_i, we first estimate $\eta(x_i) \stackrel{\text{def}}{=} \mathbb{P}[y_i = 1|x_i]$ using $h_{A,t-1}$, and then use the entropy of the label distribution of x_i to calculate the probability of querying x_i. The estimate of $\eta(\cdot)$ in round t depends both on the current active learner $h_{A,t-1}$, and the loss function. In general it is not possible to estimate $\eta(\cdot)$ with arbitrary convex loss functions. However it has been shown by Zhang (2004) that the squared, logistic and exponential losses tend to estimate the underlying conditional distribution $\eta(\cdot)$. Steps 4 of the algorithm 1 depends on the loss function, $\phi(\cdot)$, and calculates the conditional probability of the point being labeled +1. If we use the logistic loss i.e $\phi(yz) = \ln(1+\exp(-yz))$ then $\hat{\eta}_t(x) = \frac{1}{1+\exp(-yh_{A,t-1}^T x)}$. In case of squared loss $\hat{\eta}_t(x) = \min\{\max\{0, w_{A,t-1}^T x\}, 1\}$. Step 5 calculates the sampling probability of point x_i in round t, via the entropy, $H(p) \stackrel{\text{def}}{=} -p\ln(p) - (1-p)\ln(1-p)$, of the conditional probability calculated in step 4 to generate the sampling distribution over the pool $\mathcal{P}$. Notice that the minimum probability of sampling a point in round t is $p_{\min}^t$, which is calculated in step 3 as $\frac{1}{nt^\kappa}$. The role of κ is to trade-off exploration and exploitation. If $\kappa = 0$ then in each round we uniformly sample over the entire pool, and this is nothing but pure exploration. For larger κ, the minimum probability at every point is small, yet non-zero, and this can be seen as exploiting more often than exploring. Theorem 3 suggests $\kappa = 1/2$, but experimentally we noticed that one can take κ as large as 1. By design UPAL might re-query points. An alternate strategy is to not allow re-querying of points. However the importance weighted risk may not be an unbiased estimator of the true risk in such a case. Hence in order to retain the unbiasedness property we allow re-querying in UPAL.

2.1 The case of squared loss

It is interesting to look at the behaviour of UPAL in the case of squared loss where $\phi(yh^Tx) = (1-yh^Tx)^2$. For the rest of the paper we shall denote by h_A the hypothesis returned by UPAL at the end of T rounds. We now show that the prediction of h_A on any x is simply the exponentially weighted average of predictions of all h in $\mathcal{H}$.

Algorithm 1 UPAL (Input: $\mathcal{P} = \{x_1, \ldots, x_n, \}$, Loss function $\phi(\cdot)$, Budget B, Labeling Oracle $\mathcal{O}$,$\kappa \geq 0$)

1. Set num_unique_queries=0, $h_{A,0} = 0$, $t = 1$.
while num_unique_queries $\leq B$ **do**
  2. Set $Q_i^t = 0$ for all $i = 1, \ldots, n$.
  for $x_1, \ldots, x_n \in \mathcal{P}$ **do**
    3. Set $p_{\min}^t = \frac{1}{nt^\kappa}$.
    4. Calculate $\hat{\eta}_t(x_i) = \mathbb{P}[y = +1|x_i, h_{A,t-1}]$.
    5. $p_i^t \stackrel{\text{def}}{=} p_{\min}^t + (1 - np_{\min}^t)\frac{H(\hat{\eta}_t(x_i))}{\sum_{j=1}^n H(\hat{\eta}_t(x_j))}$.
  end for
  6. Sample a point (say x_j) from $p^t(\cdot)$.
  if x_j was queried previously **then**
    7. Reuse its previously queried label y_j.
  else
    8. Query oracle $\mathcal{O}$ for its label y_j.
    9. num_unique_queries $\leftarrow$ num_unique_queries+1.
  end if
  10. Set $Q_j^t = 1$.
  11. Solve the optimization problem: $h_{A,t} = \arg\min_{h\in\mathcal{H}} \sum_{i=1}^n \sum_{\tau=1}^t \frac{Q_i^\tau}{p_i^\tau}\phi(y_i h^T x_i)$.
  12. $t \leftarrow t+1$.
end while
13. Return $h_A \stackrel{\text{def}}{=} h_{A,t}$

Theorem 2. *Let*

$$z_i \stackrel{\text{def}}{=} \sum_{t=1}^T \frac{Q_i^t}{p_i^t} \qquad \hat{\Sigma}_z \stackrel{\text{def}}{=} \sum_{i=1}^n z_i x_i x_i^T$$

$$v_z \stackrel{\text{def}}{=} \sum_{i=1}^n z_i y_i x_i \qquad c \stackrel{\text{def}}{=} \sum_{i=1}^n z_i.$$

Define $w \in \mathbb{R}^d$ as

$$w = \frac{\int_{\mathbb{R}^d} \exp(-\hat{L}_T(h)) h \ \mathrm{d}h}{\int_{\mathbb{R}^d} \exp(-\hat{L}_T(h)) \ \mathrm{d}h} \tag{1}$$

Assuming M is invertible we have for any $x_0 \in \mathbb{R}^d$, $w^T x_0 = h_A^T x_0$.

Proof. By elementary linear algebra one can establish that

$$h_A = \hat{\Sigma}_z^{-1} v_z \tag{2}$$

$$\hat{L}_T(h) = (h - \hat{\Sigma}_z^{-1} v_z)\hat{\Sigma}_z(h - \hat{\Sigma}_z^{-1} v_z) \tag{3}$$

Using standard integrals we get

$$Z \stackrel{\text{def}}{=} \int_{\mathbb{R}^d} \exp(-\hat{L}_T(h)) \ \mathrm{d}h = \exp(-c - v_z^T \hat{\Sigma}_z^{-1} v_z)\sqrt{\pi^d}\sqrt{\det(\hat{\Sigma}_z^{-1})}. \tag{4}$$

In order to calculate $w^T x_0$, it is now enough to calculate the integral

$$I \stackrel{\text{def}}{=} \int_{\mathbb{R}^d} \exp(-\hat{L}_T(h))\ h^T x_0\ \mathrm{d}w.$$

To solve this integral we proceed as follows. Define $I_1 = \int_{\mathbb{R}^d} \exp(-\hat{L}_T(h))\ h^T x_0\ \mathrm{d}h$. By simple algebra we get

$$I = \int_{\mathbb{R}^d} \exp(-w^T \hat{\Sigma}_z w + 2w^T v_z - c)\ w^T x_0\ \mathrm{d}w \quad (5)$$

$$= \exp(-c - v_z^T \hat{\Sigma}_z^{-1} v_z) I_1 \quad (6)$$

Let $a = h - \hat{\Sigma}_z^{-1} v_z$. We then get

$$\begin{aligned} I_1 &= \int_{\mathbb{R}^d} h^T x_0 \exp\left(-(h - \hat{\Sigma}_z^{-1} v_z)\hat{\Sigma}_z (h - \hat{\Sigma}_z^{-1} v_z)\right)\ \mathrm{d}h \\ &= \int_{\mathbb{R}^d} (a^T x_0 + v_z^T \hat{\Sigma}_z^{-1} x_0) \exp(-a^T \hat{\Sigma}_z a)\ \mathrm{d}a \\ &= \underbrace{\int_{\mathbb{R}^d} (a^T x_0) \exp(-a^T \hat{\Sigma}_z a)\ \mathrm{d}a}_{I_2} + \\ &\qquad \underbrace{\int_{\mathbb{R}^d} v_z^T \hat{\Sigma}_z^{-1} x_0 \exp(-a^T \hat{\Sigma}_z a)\ \mathrm{d}a}_{I_3} \end{aligned} \quad (7)$$

Clearly I_2 being the integrand of an odd function over the entire space is equal to 0. To calculate I_3 we shall substitute $\hat{\Sigma}_z = SS^T$, where $S \succ 0$. Such a decomposition is possible since $\hat{\Sigma}_z \succ 0$. Now define $z = S^T a$. We get

$$I_3 = v_z^T \hat{\Sigma}_z^{-1} x_0 \int \exp(-z^T z)\ \det(S^{-1})\ \mathrm{d}z \quad (8)$$

$$= v_z^T \hat{\Sigma}_z^{-1} x_0 \det(S^{-1}) \sqrt{\pi^d} \quad (9)$$

Using equations (6, 7, 9) we get

$$I = (\sqrt{\pi})^d v_z^T \hat{\Sigma}_z^{-1} x_0\ \det(S^{-1}) \exp(-c - v_z^T \hat{\Sigma}_z^{-1} v_z). \quad (10)$$

Hence we get

$$w^T x_0 = v_z^T \hat{\Sigma}_z^{-1} x_0 \frac{\det(S^{-1})}{\sqrt{\det(\hat{\Sigma}_z^{-1})}} = v_z^T \hat{\Sigma}_z^{-1} x_0 = h_A^T x_0,$$

where the penultimate equality follows from the fact that $\det(\hat{\Sigma}_z^{-1}) = 1/\det(\hat{\Sigma}_z) = 1/(\det(SS^T)) = 1/(\det(S))^2$, and the last equality follows from equation 2. □

Theorem 2 is instructive. It tells us that assuming that the matrix $\hat{\Sigma}_z$ is invertible, h_A is the same as an exponentially weighted average of all the hypothesis in $\mathcal{H}$. Hence one can view UPAL as learning with expert advice, where each individual hypothesis $h \in \mathcal{H}$ is an expert, and the exponential of $\hat{L}_T$ is used to weigh the hypothesis in $\mathcal{H}$. Such forecasters have been commonly used in learning with expert advice. This also lends a different interpretation for UPAL. UPAL prunes the hypothesis space in a soft way via the exponential weighting scheme. The hypothesis that has suffered more cumulative loss get lesser weight, while the ones that has suffered lesser cumulative loss get more weight.

3 Consistency of UPAL

It is natural to ask if UPAL is consistent. That is will UPAL do as well as the optimal hypothesis in $\mathcal{H}$ as $n \to \infty, T \to \infty$? We asnwer this question in affirmative for a restricted setting using the squared loss. Let us denote by h_A the hypothesis obtained after we have run UPAL for T rounds. Also let us suppose that $y \in [-1, +1]$. This can be seen as the oracle returning the conditional expectation $\mathbb{E}[y|x]$ instead of just the sign of $\mathbb{E}[y|x]$. We shall make the following assumptions 1) **[A0]** $\Sigma \stackrel{\text{def}}{=} \mathbb{E}[xx^T]$ is invertible. 2) **[A1]** $||x_i|| \leq B$ a.s. 3) **[A2]** $y = \beta^T x$ a.s. A0 is required to guarantee that there is a unique minimizer of the expected squared loss. A1 is just a boundedness assumption of the input domain. The strongest assumption is A2. While the assumption of linear model may be valid in kernel spaces, the assumption that the observations are non-noisy is indeed somewhat restrictive. Our current proof is only for the squared loss. The motivation for using squared loss is that it leads to closed form solution for h_A, which can then be elegantly analyzed using results from random matrix theory (Hsu et al., 2011a,b; Tropp, 2010). It may be possible to extend these results to other loss functions such as the logistic loss, or exponential loss using results from empirical process theory (van de Geer, 2000), and the idea of self-concordant functions (Bach, 2010)

Our main result is that under assumptions A0-A3, given enough data, and if UPAL with the squared loss, and $\kappa = 1/2$ is run for enough rounds, then with high probability over the sample and the randomness in sampling $h_A = \beta$.

Theorem 3. *Suppose assumptions A0-A2 hold. Then for $T \geq T_{0,\delta}$, $n \geq \max\{n_{0,\delta}, n_{1,\delta}\}$, $\kappa = 1/2$ with probability at least $1 - 5\delta$, UPAL recovers the vector β.*

The rough proof sketch is as follows. We first establish in lemma 1 that conditioned on the invertibility of matrices $\hat{\Sigma}_z, \hat{\Sigma}$ ($\hat{\Sigma}$ is the empirical covariance matrix) the hypothesis h_A returned by UPAL is β. Once we have

established this simple result, we establish conditions for the invertibility of matrices $\hat{\Sigma}_z, \hat{\Sigma}$ in lemmas 2, 3. We will require the following notation in addition to what has been used in theorem 2

$$n_{1,\delta} \stackrel{\text{def}}{=} \frac{7200d^2B^4}{\lambda^2_{\min}(\Sigma)}(d\ln(5)+\ln(2/\delta)) \; n_{0,\delta} \stackrel{\text{def}}{=} \frac{8B^2\ln(d/\delta)}{\lambda_{\min}(\Sigma)}$$

$$T_{0,\delta} \stackrel{\text{def}}{=} \frac{324B^8}{(\lambda_{\min}(\Sigma))^4} + \frac{18\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}\ln(d/\delta) \;\; \hat{\Sigma} \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$$

$\mathbb{E}_t[\cdot] \stackrel{\text{def}}{=} \mathbb{E}_{Q^t_{1:n}|Q^{1:t-1}_{1:n},\mathcal{D}_n}[\cdot]$.

Lemma 1. *Suppose assumptions A0-A2 hold. Then conditioned on the invertibility of matriecs $\hat{\Sigma}_z, \hat{\Sigma}$, for $T \geq T_{0,\delta}$, $n \geq \max\{n_{0,\delta}, n_{1,\delta}\}$ we have $h_A = \beta$.*

Proof.

$$h_A = \hat{\Sigma}_z^{-1}v_z = \hat{\Sigma}_z^{-1}\sum_{i=1}^{n} z_i y_i x_i = \hat{\Sigma}_z^{-1}\hat{\Sigma}_z\beta = \beta,$$

where in the first step we used the fact that h_A is the active learner outputted by the algorithm after T rounds, and in the third equality we used assumption A2. □

Lemma 1 makes use of the assumption that both $\hat{\Sigma}_z^{-1}, \hat{\Sigma}^{-1}$ are well defined. The next lemma establishes conditions on the invertibility of these matrices. The key tool is to use results regarding spectra of random matrices. In particular we shall be using the matrix Bernstein bound and the matrix Chernoff bound. These results have been stated in the appendix.

Lemma 2. *With probability at least $1-\delta$ each the following two inequalities hold*

1. $\lambda_{min}(\hat{\Sigma}) \geq \frac{1}{2}\lambda_{min}(\Sigma) > 0$ *for* $n \geq n_{0,\delta}$.
2. $\lambda_{max}(\hat{\Sigma}) \leq \frac{3}{2}\lambda_{max}(\Sigma)$ *for* $n \geq n_{1,\delta}$.

Proof. Let $J \stackrel{\text{def}}{=} \sum_{i=1}^{n}\Sigma^{-1/2}x_i x_i^T\Sigma^{-1/2}$. From the definition of matrix norms we get

$$||\Sigma^{-1/2}x_i|| \leq ||\Sigma^{-1/2}||\;||x_i|| \leq \frac{B}{\sqrt{\lambda_{\min}(\Sigma)}}.$$

To prove the first part we shall now use the matrix Chernoff inequality (Theorem 5) where $b \stackrel{\text{def}}{=} \frac{B}{\sqrt{\lambda_{\min}(\Sigma)}}$. We then get with probability at least $1-\delta$

$$\lambda_{\min}(J/n) \geq 1 - \sqrt{\frac{2B^2\ln(d/\delta)}{n\lambda_{\min}(\Sigma)}} \geq 1/2 \qquad (11)$$

for $n \geq n_{0,\delta}$. Now by definition $J = n\Sigma^{-1/2}\hat{\Sigma}\Sigma^{-1/2}$.

$$\lambda_{\min}(\hat{\Sigma}) = \frac{1}{\lambda_{\max}(\hat{\Sigma}^{-1})} = \frac{1}{n\lambda_{\max}(\Sigma^{-1/2}J^{-1}\Sigma^{-1/2})} = \frac{1}{n||\Sigma^{-1/2}J^{-1}\Sigma^{-1/2}||} \geq \frac{1}{n||\Sigma^{-1/2}||\;||J^{-1}||\;||\Sigma^{-1/2}||} = \frac{1}{n}\lambda_{\min}(\Sigma)\lambda_{\min}(J) \geq \frac{\lambda_{\min}(\Sigma)}{2}, \qquad (12)$$

where in the last line we used equation 11. Proof of the second part is similar but we now instead use the matrix Bernstein inequality (Theorem 4). □

We are now ready to establish conditions for the invertibility of $\hat{\Sigma}_z$.

Lemma 3. *For $T \geq T_{0,\delta}$, and $\kappa = 1/2$, with probability at least $1-4\delta$ we have $\lambda_{min}(\hat{\Sigma}_z) \geq nT\lambda_{min}(\Sigma)/12 > 0$, and hence $\hat{\Sigma}_z$ is invertible.*

Proof. The idea is to use the matrix Bernstein bound (theorem 4) to get a lower bound on $\lambda_{\min}(\hat{\Sigma}_z)$. Let $M'_t \stackrel{\text{def}}{=} \sum_{i=1}^{n}\frac{Q_i^t}{p_i^t}x_i x_i^T$, so that $\hat{\Sigma}_z = \sum_{t=1}^{T}M'_t$. Now $\mathbb{E}_t M'_t = n\hat{\Sigma}$. Define $R'_t \stackrel{\text{def}}{=} n\hat{\Sigma} - M'_t$, so that $\mathbb{E}_t R'_t = 0$. We shall apply the matrix Bernstein inequality to the random matrix $\sum R'_t$. To do so we need upper bounds on $\lambda_{\max}(R'_t)$ and $\lambda_{\max}(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_t R'^2_t)$. Let $n \geq n_{1,\delta}$. Using lemma 2 we get with probability at least $1-\delta$

$$\lambda_{\max}(R'_t) = \lambda_{\max}(n\hat{\Sigma} - M'_t) \leq \lambda_{\max}(n\hat{\Sigma}) \leq \frac{3n\lambda_{\max}(\Sigma)}{2} \stackrel{\text{def}}{=} b_2. \qquad (13)$$

$$\lambda_{\max}\left[\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_t R'^2_t\right] = \frac{1}{T}\lambda_{\max}[\sum_{t=1}^{T}\mathbb{E}_t(n\hat{\Sigma} - M'_t)^2] \qquad (14)$$

$$= \frac{1}{T}\lambda_{\max}(-n^2T\hat{\Sigma}^2 + \sum_{t=1}^{T}\mathbb{E}_t\sum_{i=1}^{n}\frac{Q_i^t}{(p_i^t)^2}(x_i x_i^T)^2) \qquad (15)$$

$$= \frac{1}{T}\lambda_{\max}(-n^2T\hat{\Sigma}^2 + \sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{p_i^t}(x_i x_i^T)^2) \qquad (16)$$

$$\leq \frac{1}{T}\lambda_{\max}(\sum_{i=1}^{n}\sum_{t=1}^{T}\frac{1}{p_i^t}(x_i x_i^T)^2) \qquad (17)$$

$$\leq n\sqrt{T}\lambda_{\max}(\sum_{i=1}^{n}(x_i x_i^T)^2) \qquad (18)$$

$$\leq n\sqrt{T}\sum_{i=1}^{n}\lambda^2_{\max}(x_i x_i^T) \qquad (19)$$

$$\leq n^2\sqrt{T}B^4 \stackrel{\text{def}}{=} \sigma_2^2. \qquad (20)$$

Equation 15 follows from equation 14 by the definition of M'_t and the fact that in any given round only one

point is queried, i.e for a given t and $i \neq j$ we get $Q_i^t Q_j^t = 0$. Equation 16 follows from 15 by using the fact that $E_t Q_i^t = p_i^t$. Equation 17 follows from 16 by Weyl's inequality and the fact that $\hat{\Sigma} \succeq 0$. To obtain 18 from 17 we substituted $p_{\min}^t \stackrel{\text{def}}{=} \frac{1}{n\sqrt{t}}$ in place of p_i^t. Finally the remaining set of inequalities follow because of A1, and the fact that if p is a vector then $\lambda_{\max}(pp^T) = ||p||^2$.

Using theorem 4 we get with probability at least $1-\delta$,

$$\lambda_{\max}(\frac{1}{T}\sum_{t=1}^{T} R_t') \leq \sqrt{\frac{2\sigma_2^2 \ln(d/\delta)}{T}} + \frac{b_2 \ln(d/\delta)}{T}$$

$$\implies \lambda_{\max}(n\hat{\Sigma} - \frac{1}{T}\sum_{t=1}^{T} M_t') \leq \sqrt{\frac{2\sigma_2^2 \ln(d/\delta)}{T}} + \frac{b_2 \ln(d/\delta)}{T}$$

$$\implies \lambda_{\min}(n\hat{\Sigma}) - \frac{1}{T}\lambda_{\min}\left(\sum_{t=1}^{T} M_t'\right) \leq \sqrt{\frac{2\sigma_2^2 \ln(d/\delta)}{T}} + \frac{b_2 \ln(d/\delta)}{T} \quad (21)$$

Rearranging the inequality and substituting for σ_2, b_2 as calculated in equations 13, 20

$$\lambda_{\min}(\sum_{t=1}^{T} M_t') \geq nT\lambda_{\min}(\hat{\Sigma}) - \sqrt{2T^{3/2}n^2B^4 \ln(d/\delta)} - \frac{3n\lambda_{\max}(\Sigma)\ln(d/\delta)}{2}. \quad (22)$$

By union bound the above stochastic inequality holds with probability at least $1-3\delta$. Finally using lemma 2 to stochastically lower bound the quantity $\lambda_{\min}(\hat{\Sigma})$ by $\lambda_{\min}(\Sigma)/2$, and applying union bound once again we get the desired result. □

Proof of theorem 3. For $n \geq n_{0,\delta}$ from lemma 2, with probability $1-\delta$, $\hat{\Sigma}$ is invertible. For $T \geq T_{0,\delta}, n \geq n_{1,\delta}$ the matrix $\hat{\Sigma}_z$ becomes invertible with probability atleast $1-4\delta$. Conditioned on the invertibility of $\hat{\Sigma}, \hat{\Sigma}_z$ from lemma 1 we can recover β exactly. Summing up all the failure probabilities using union bound we get the desired result. □

4 Related Work

A variety of pool based AL algorithms have been proposed in the literature employing various query strategies. However, none of them use unbiased estimates of the risk. One of the simplest strategy for AL is uncertainty sampling, where the active learner queries the point whose label it is most uncertain about. This strategy has been popular in text classification (Lewis and Gale, 1994), and information extraction (Settles and Craven, 2008). Usually the uncertainty in the label is calculated using certain information-theoretic criteria such as entropy, or variance of the label distribution. While uncertainty sampling has mostly been used in a probabilistic setting, AL algorithms which learn non-probabilistic classifiers using uncertainty sampling have also been proposed. Tong et al. (2001) proposed an algorithm in this framework where they query the point closest to the current svm hyperplane. Seung et al. (1992) introduced the query-by-committee (QBC) framework where a committee of potential models, which all agree on the currently labeled data is maintained and, the point where most committee members disagree is considered for querying. In order to design a committee in the QBC framework, algorithms such as query-by-boosting, and query-by-bagging in the discriminative setting (Abe and Mamitsuka, 1998), sampling from a Dirichlet distribution over model parameters in the generative setting (McCallum and Nigam, 1998) have been proposed. Other frameworks include querying the point, which causes the maximum expected reduction in error (Zhu et al., 2003; Guo and Greiner, 2007), variance reducing query strategies such as the ones based on optimal design (Flaherty et al., 2005; Zhang and Oles, 2000). A very thorough literature survey of different active learning algorithms has been done by Settles (2009). AL algorithms that are consistent and have provable label complexity have been proposed for the agnostic setting for the 0-1 loss in recent years (Dasgupta et al., 2007; Beygelzimer et al., 2009). The IWAL framework introduced in Beygelzimer et al. (2009) was the first AL algorithm with guarantees for general loss functions. However the authors were unable to provide non-trivial label complexity guarantees for the hinge loss, and the squared loss.

UPAL at least for squared losses can be seen as using a QBC based querying strategy where the committee is the entire hypothesis space, and the disagreement among the committee members is calculated using an exponential weighting scheme. However unlike previously proposed committees our committee is an infinite set, and the choice of the point to be queried is randomized.

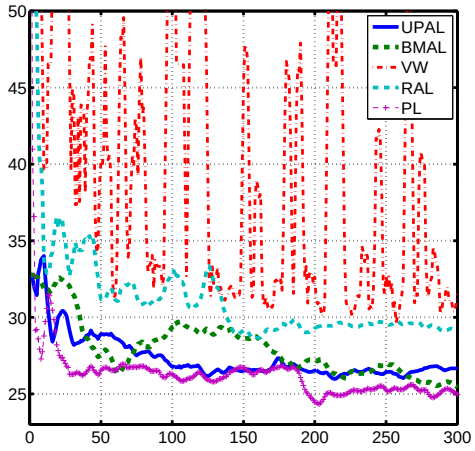
5 Experimental results

We implemented UPAL, along with the standard passive learning (PL) algorithm, and a variant of UPAL called RAL (in short for random active learning), all using logistic loss, in MATLAB. The choice of logistic loss was motivated by the fact that BMAL was designed for logistic loss. Our MATLAB codes were vectorized to the maximum possible extent so as to be as efficient as possible. RAL is similar to UPAL, but

in each round samples a point uniformly at random from the currently unqueried pool. However it does not use importance weights to calculate an estimate of the risk of the classifier. The purpose of implementing RAL was to demonstrate the potential effect of using unbiased estimators, and to check if the strategy of randomly querying points helps in active learning.

We also implemented a batch mode active learning algorithm introduced by Hoi et al. (2006) which, we shall call as BMAL. Hoi et al. in their paper showed superior empirical performance of BMAL over other competing pool based active learning algorithms, and this is the primary motivation for choosing BMAL as a competitor pool AL algorithm in this paper. BMAL like UPAL also proceeds in rounds and in each iteration selects k examples by minimizing the Fisher information ratio between the current unqueried pool and the queried pool. However a point once queried by BMAL is never requeried. In order to tackle the high computational complexity of optimally choosing a set of k points in each round, the authors suggested a monotonic submodular approximation to the original Fisher ratio objective, which is then optimized by a greedy algorithm. At the start of round $t+1$ when, BMAL has already queried t points in the previous rounds, in order to decide which point to query next, BMAL has to calculate for each potential new query a dot product with all the remaining unqueried points. Such a calculation done for all possible potential new queries takes $O(n^2t)$ time. Hence if our budget is B, then the total computational complexity of BMAL is $O(n^2B^2)$. Note that this calculation does not take into account the complexity of solving an optimization problem in each round after having queried a point. In order to further reduce the computational complexity of BMAL in each round we further restrict our search, for the next query, to a small subsample of the current set of unqueried points. In our experiments the size of the subsample is taken to be 300. In order to avoid numerical problems we implemented a regularized version of UPAL where the term $\lambda||w||^2$ was added to the optimization problem shown in step 11 of Algorithm 1. The value of λ is allowed to change as per the current importance weight of the pool. The optimal value of C in VW [1] was chosen via a 5 fold cross-validation, and by eyeballing for the value of C that gave the best cost-accuracy trade-off. Figure 1 shows the performance of all the algorithms on the first 300 queried points.

[1]The parameters initial_t, l were set to a default value of 10 for all of our experiments.



(d) Whitewine

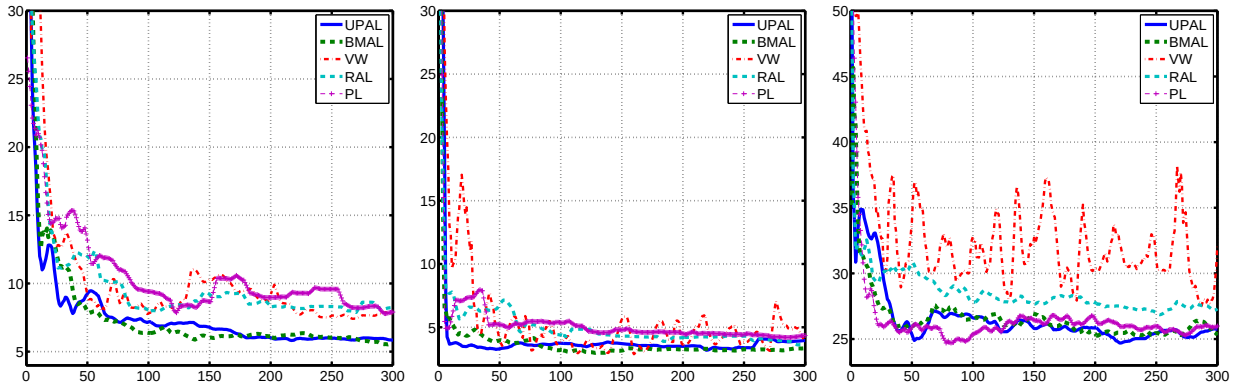
Figure 1: Empirical performance of passive and active learning algorithms.The x-axis represents the number of points queried, and the y-axis represents the test error of the classifier. The value of κ was set to 1.

Sample size	UPAL		BMAL	
	Time	Error	Time	Error
1200	65	7.27	60	5.67
2400	100	6.25	152	6.05
4800	159	6.83	295	6.25
10000	478	5.85	643.17	5.85

Table 1: Comparison of UPAL and BMAL on MNIST data-set of varying training sizes, and with the budget being fixed at 300. The error rate is in percentage, and the time is in seconds.

Budget	UPAL		BMAL		Speedup
	Time	Error	Time	Error	
500	859	5.79	1973	5.33	2.3
1000	1919	6.43	7505	5.70	3.9
2000	4676	5.82	32186	5.59	6.9

Table 2: Comparison of UPAL on the MNIST (3 Vs 5) dataset, of size 10000 for varying budget size. All the times are in seconds unless stated, and error rates in percentage.



(a) MNIST (3 vs 5) (b) Statlog (c) Abalone

On the MNIST dataset, on an average, the performance of BMAL is very similar to UPAL, and there is a noticeable gap in the performance of BMAL and UPAL over PL, VW and RAL. Similar results were also seen in the case of Statlog dataset, though towards the end the performance of UPAL slightly worsens when compared to BMAL. However UPAL is still better than PL, VW, and RAL.

Active learning is not always helpful and the success story of AL depends on the match between the marginal distribution and the hypothesis class. This is clearly reflected in Abalone where the performance of PL is better than UPAL at least in the initial stages and is never significantly worse. UPAL is uniformly better than BMAL, though the difference in error rates is not significant. However the performance of RAL, VW are significantly worse. Similar results were also seen in the case of Whitewine dataset, where PL outperforms all AL algorithms. UPAL is better than BMAL most of the times. Even here one can witness a huge gap in the performance of VW and RAL over others.

One can conclude that VW though is computationally efficient has higher error rate for the same number of queries. The uniformly poor performance of RAL signifies that querying uniformly at random does not help. On the whole UPAL and BMAL perform equally well, and we show via our next set of experiments that UPAL has significantly better scalability, especially when one has a relatively large budget B.

5.1 Scalability results

Each round of UPAL takes $O(n)$ plus the time to solve the optimization problem shown in step 11 in Algorithm 1. A similar optimization problem is also solved in the BMAL problem. If the cost of solving this optimization problem in step t is $c_{opt,t}$, then the complexity of UPAL is $O(nT + \sum_{t=1}^{T} c_{opt,t})$. While BMAL takes $O(n^2B^2 + \sum_{t=1}^{T} c'_{t,opt})$ where $c'_{t,opt}$ is the complexity of solving the optimization problem in BMAL in round t. For the approximate implementation of BMAL that we described if the subsample size is $|S|$, then the complexity is $O(|S|^2B^2 + \sum_{t=1}^{T} c'_{t,opt})$.

In our first set of experiments we fixed the budget $B = 300$, and calculated the error rate and the combined training and testing time of both BMAL and UPAL for varying sizes of the training set. All the experiments were performed on the MNIST dataset. Table 1 shows that with increasing sample size UPAL tends to be more efficient than BMAL, though the gain in speed that we observed was at most a factor of 1.8.

In the second set of scalability experiments (Table 2), we studied the effect of increasing budget. We found out that with increasing budget size the speedup of UPAL over BMAL increases. In particular when the *budget was 2000, UPAL is arpproximately 7 times faster than BMAL.* All our experiments were run on a dual core machine with 3 GB memory.

6 Conclusions and Discussion

In this paper we proposed the first unbiased pool based active learning algorithm, and showed its good empirical performance and its ability to scale with higher budget constraints. Since the submission of this paper we have been able to extend our consistency result to the noisy setting, details of which can be found in Ganti and Gray (2011). We believe our consistency results can be improved. An important question to solve is to give explicit upper bounds on the number of unique queries made by UPAL in T rounds of the algorithm. This will allow us to compare pool based AL algorithms and stream based AL algorithms theoretically.

References

N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *ICML*, 1998.

F. Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4:384–414, 2010.

E.B. Baum and K. Lang. Query learning can work poorly when a human oracle is used. In *IJCNN*, 1992.

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

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge Univ Press, 2006.

W. Chu, M. Zinkevich, L. Li, A. Thomas, and B. Tseng. Unbiased online active learning in data streams. In *SIGKDD*, 2011.

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

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

Patrick Flaherty, Michael I. Jordan, and Adam P. Arkin. Robust design of biological experiments. In *Neural Information Processing Systems*, 2005.

R. Ganti and A. Gray. Upal: Unbiased pool based active learning. *Arxiv preprint arXiv:1111.1784*, 2011.

Y. Guo and R. Greiner. Optimistic active learning using mutual information. In *IJCAI*, 2007.

S.C.H. Hoi, R. Jin, J. Zhu, and M.R. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, 2006.

R.A. Horn and C.R. Johnson. *Matrix analysis*. Cambridge Univ Press, 1990.

D. Hsu, S.M. Kakade, and T. Zhang. An analysis of random design linear regression. *Arxiv preprint arXiv:1106.2363*, 2011a.

D. Hsu, S.M. Kakade, and T. Zhang. Dimension-free tail inequalities for sums of random matrices. *Arxiv preprint arXiv:1104.1672*, 2011b.

J. Langford, L. Li, A. Strehl, D. Hsu, N. Karampatziakis, and M. Hoffman. Vowpal wabbit, 2011.

D.D. Lewis and W.A. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, 1994.

A.K. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *ICML*, 1998.

J. Quinonero, M. Sugiama, A. Schwaighofer, and N.D. Lawrence. Dataset shift in machine learning, 2008.

B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *EMNLP*, 2008.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *COLT*, pages 287–294. ACM, 1992.

S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, 2001.

J.A. Tropp. User-friendly tail bounds for sums of random matrices. *Arxiv preprint arXiv:1004.4389*, 2010.

Sara van de Geer. Empirical processes in m-estimation. 2000.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32(1), 2004.

T. Zhang and F. Oles. The value of unlabeled data for classification problems. In *ICML*, 2000.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, 2003.

A Some results from random matrix theory

Theorem 4. *(Matrix Bernstein bound Hsu et al. (2011a)) Let $M_1 \ldots, M_n$ be symmetric valued random matrices. Suppose there exist $\bar{b}, \bar{\sigma}$ such that for all $i = 1, \ldots, n$*

$$\mathbb{E}_i[M_i] = 0$$
$$\lambda_{max}(M_i) \leq \bar{b}$$
$$\lambda_{max}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_i[M_i^2]\right) \leq \bar{\sigma}^2,$$

almost surely, then

$$\mathbb{P}\left[\lambda_{max}\left(\frac{1}{n}\sum_{i=1}^{n}M_i\right) > \sqrt{\frac{2\bar{\sigma}^2\ln(d/\delta)}{n}} + \frac{\bar{b}\ln(d/\delta)}{3n}\right] \leq \delta.$$

A dimension free version of the above inequality was proved in Hsu et al. (2011b). Such dimension free inequalities are especially useful in infinite dimensional spaces. However since we are working in a finite dimensional space we shall stick to the non-dimension free version.

Theorem 5. *(Matrix chernoff bound; Tropp (2010); Hsu et al. (2011a)) Let $v_1, \dots v_n$ be random vectors such that, for some $b \geq 0$*

$$\mathbb{E}[||v_i||^2 | v_1, \dots, v_{i-1}] \geq 1, \text{and } ||v_i|| \leq b,$$

for all $i = 1, \dots, n$, almost surely. For all $\delta \in (0, 1)$,

$$\mathbb{P}\left[\lambda_{min}\left(\frac{1}{n}\sum_{i=1}^{n} v_i v_i^T\right) < 1 - \sqrt{\frac{2b^2}{n}\ln(d/\delta)}\right] \leq \delta. \tag{23}$$

Theorem 6. *Let A, B be positive semidefinite matrices. Then*

$$\lambda_{max}(A)+\lambda_{min}(B) \leq \lambda_{max}(A+B) \leq \lambda_{max}(A)+\lambda_{max}(B)$$

The above inequalities are known as Weyl's inequalities (see Horn and Johnson, 1990, chap. 3)