Active Learning with Importance Sampling

Muni Sreenivas Pydi
Department of Electrical & Computer Engineering
University of Wisconsin-Madison
Madison, WI 53706
pydi@wisc.edu

Vishnu Lokhande
Department of Computer Science
University of Wisconsin-Madison
Madison, WI 53706
lokhande@cs.wisc.edu

Abstract

We consider an active learning setting where the algorithm has access to a large pool of unlabeled data and a small pool of labeled data. In each iteration, the algorithm chooses few unlabeled data points and obtains their labels from an oracle. In this paper, we propose a probabilistic querying procedure to choose the points. We prove that the procedure is optimal in terms of minimizing the true error of the algorithm. Using measure concentration, we prove upper bounds on the true error for the proposed scheme.

1 Introduction

Active learning is an important machine learning paradigm with a rich class of problems and mature literature Prince [2004], Settles [2012], Hanneke et al. [2014]. In may applications, the active learning algorithm has access to a large pool of unlabeled data and access to an oracle that provides labels to any given data point. Querying the oracle for the label comes at a cost, either computational or monetary. Hence, a key objective for the algorithm in this setting is to choose the data points to be labeled by the oracle "wisely" so that the test accuracy increases fast as more and more data points are labeled.

Importance sampling is a popular statistical technique used in several machine learning algorithms. For instance, importance sampling is widely used for fast convergence in Monte Carlo based methods Doucet et al. [2001] and stochastic optimization Zhao and Zhang [2015]. In this paper, we propose a probabilistic querying procedure motivated from importance sampling literature to choose the points to be labeled by the oracle in an active learning setting.

2 Setting

We consider a binary classification problem where there is a large pool of $i.i.d.$ data points $\{x_i, y_i\}_{i\in[n]} \sim \mathcal{Z}$ drawn from an underlying probability distribution $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, out of which, a small pool of data points $S^0 = \{s^0(j) \in [n]\}_{j\in[m]}$ are labeled at the start. This initial labeled pool is also $i.i.d.$ on the underlying probability distribution.

At the start, an active algorithm $\mathcal{A}^0$ has access to all the data $T = \{x_i\}_{i\in[n]}$ in the large pool and the labels $\{y_{s^0(j)}\}_{j\in[m]}$ for the data in the small pool S^0. For the next iteration, the sampling procedure defined in «» is used to select a small subset, $S^1 = \{s^1(j) \in [n]\}_{j\in[m]}$ of size m, for which the labels are requested from the oracle. The algorithm $\mathcal{A}^1$ now has access to the labels for the data points in $S^0 \cup S^1$ and subsequently requests labels for yet another small subset S^2.

At the end of t iterations, algorithm $\mathcal{A}^t$ has access to labels for the data points in $S^0 \cup \ldots \cup S^t$. Let $l(x_i, y_i; \mathcal{A}^t)$ be the loss function for the data point (x_i, y_i) evaluated on the algorithm A^t. Let $n_t = n - tm$ denote the number of unlabeled points after t iterations. Let $U^t = \{u^t(j) \in [n]\}_{j\in[n_t]}$ denote the set of unlabeled points after t iterations.

3 Problem Formulation

Let $\{Q_i^t\}_{i\in[n_t]}$ denote a set of Bernoulli random variables such that $\mathbb{P}(Q_i^t = 1) = p_i^t$. A label for the data point x_i is requested from the oracle only when $Q_i^t = 1$. Hence, $\{p^t_{u^t(j)}\}_{i\in[n_t]}$ denote the probabilities of querying the labels for the unlabeled points in U^t at the end of t iterations. For convenience, we write p^t to denote this probability distribution. The objective of this paper is to derive an expression for this optimal querying probability distribution that minimizes the true error rate $\mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t)]$ of the algorithm $\mathcal{A}^t$.

We begin by writing an upper bound for this true error rate as follows. This decomposition of the true error into three terms is inspired from Sener and Silvio [2018]. The decomposition proposed in Sener and Silvio [2018] is identical to ours except that we used a weighted version of the queried data points to account for non-uniform sampling.

$$\mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t)] \leq \left| \mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t))] - \frac{1}{n_t} \sum_{j\in[n_t]} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \right| \tag{1}$$

$$+ \left| \frac{1}{n_t} \sum_{j\in[n_t]} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) - \frac{1}{n_t} \sum_{j\in[n_t]} \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \right| \tag{2}$$

$$+ \frac{1}{n_t} \sum_{j\in[n_t]} \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t). \tag{3}$$

The term (1) corresponds to the generalization error of the algorithm $\mathcal{A}^t$ on all the n_t unlabeled data points. Term (3) corresponds to a weighted average error for $\mathcal{A}^t$ over the data points that will be queried at the end of t iterations, i.e. the points for which $Q_i^t = 1$.Term (2) corresponds to the difference between the average error over the all the n_t unlabeled data points and the weighted average error for the queried data points.

Since $\mathbb{E}[Q_i^t] = p_i^t$, we have

$$\mathbb{E}_{Q^t}\left[\frac{1}{n_t} \sum_{j\in[n_t]} \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \right] = \frac{1}{n_t} \sum_{j\in[n_t]} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)$$

where $\mathbb{E}_{Q^t}[.]$ denotes the expectation with respect to the random variables $\{Q^t_{u_t(j)}\}_{j\in[n_t]}$. This means that the weighted average error over the points to be queried at the end of t iterations, is an *unbiased estimate* of the average loss over all the unlabeled data points in U^t. Moreover, since the data points are $i.i.d.$, the deviation in (2) can be expressed as the sum of n_t zero mean $i.i.d.$ random variables denoted by $Z_j^t = l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) - \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)$.

The variance of Z_j^t can be computed as follows.

$$\begin{aligned} Var(Z_j^t) &= Var\left(l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) - \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \right) \\ &= \frac{l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)^2}{\left(p^t_{u^t(j)}\right)^2} Var\left(Q^t_{u^t(j)}\right) \\ &= l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)^2 \left(\frac{1}{p^t_{u^t(j)}} - 1 \right) \end{aligned} \tag{4}$$

By choosing a probability distribution p^t that minimizes $\sum_{j\in[n_t]} Var(Z_j^t)$, it is possible to compute a tight upper bound on (2) using Bernstein's inequality Shalev-Shwartz and Ben-David [2014]. Given a sufficiently expressive hypothesis space and an algorithm A^t that is designed to account for weighted

loss, term (3) can be driven to zero via weighted empirical risk minimization. Hence, choosing the optimal p^t (in the sense of minimizing the variance in (4)) minimizes the true error $\mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t)]$ of the algorithm A^t.

However, the labels of the data points to be queried in iteration t are unknown to the algorithm A^t which makes computing $l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)$ impossible. To get around this problem, we use the pseudo label $\hat{y} = -sign\left(f_{\mathcal{A}}(x)\right)$ instead of the true label y, where $f_{\mathcal{A}}$ is the classifier learnt by algorithm $\mathcal{A}$. Since $l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \leq l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)$, minimizing (4) with y replaced by $\hat{y}$ still gives a valid upper bound on (2).

This idea of replacing the true label with pseudo-label and optimizing over the upperbound is borrowed from Wang and Ye [2015].

To summarize, we would like to choose a scheme for querying data points from U^t by choosing a p^t that solves the following optimization problem.

$$\min_{p_t} \sum_{j \in [n_t]} \frac{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2}{p^t_{u^t(j)}}. \tag{5}$$

4 Related Work

The closest work to our setting is Beygelzimer et al. [2009] which considers an importance weighted sampling procedure for active learning. However, the proposed algorithm does not have freedom to choose the data point to be labeled. Instead, the algorithm receives unlabeled data points in an online fashion and it has to choose whether to query the oracle for the label or not. In our setting, there is a readily available unlabeled pool of data. Moreover, the weighted sampling procedure proposed in Beygelzimer et al. [2009] is based on a version space approach over a finite hypothesis class. Our setting does not have those restrictions.

In Sener and Silvio [2018], the authors use a very similar true error decomposition as ours, but use a deterministic procedure based on coresets to choose the points to be labeled. Our approach is probabilistic, and computationally far less intensive.

5 Algorithm

Now we present a theorem that states the optimal probability distribution p^t to solve the objective function proposed in (5).

Theorem 1. *The optimal probability distribution p_t with respect to the objective function in* (5) *is given by*

$$p^t_{u^t(j)} = \frac{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)}{\sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)}. \tag{6}$$

Proof. Since p^t is a probability distribution over U^t, we have $\sum_{j \in [n_t]} p^t_{u^t(j)} = 1$. Using λ as the Lagrangian variable for this equality constraint on (5), we get the following Lagrangian for $p^t_{u^t(j)}$.

$$L\left(p^t_{u^t(j)}, \lambda\right) = \frac{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2}{p^t_{u^t(j)}} + \lambda \left(\sum_{j \in [n_t]} p^t_{u^t(j)} - 1 \right).$$

Equating $\frac{\partial L}{\partial p^t_{u^t(j)}}$ to zero, we get

$$-\frac{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2}{{p^t_{u^t(j)}}^2} + \lambda = 0$$

$$\Rightarrow p^t_{u^t(j)} = \frac{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)}{\sqrt{\lambda}}$$

Applying the equality constraint, we get (6). □

From 1, the probability of querying a data point for its label is proportional to the loss calculated on the pseudo-label at that point. Next, we give explicit expressions for p^t for specific loss functions.

Example 1.1. *For squared error loss function, the optimal querying probability in* (6) *reduces to*

$$p^t_{u^t(j)} = \frac{\left(1 + |f_{\mathcal{A}^t}(x_{u^t(j)})|\right)^2}{\sum_{j\in[n_t]} \left(1 + |f_{\mathcal{A}^t}(x_{u^t(j)})|\right)^2}.$$

Example 1.2. *For hinge loss function, the optimal querying probability in* (6) *reduces to*

$$p^t_{u^t(j)} = \frac{1 + |f_{\mathcal{A}^t}(x_{u^t(j)})|}{\sum_{j\in[n_t]} \left(1 + |f_{\mathcal{A}^t}(x_{u^t(j)})|\right)}. \tag{7}$$

The complete algorithm with the optimal importance sampling scheme is described in Algorithm 1.

Algorithm 1 Active Learning with Importance Sampling (ALIS)

Require: S^0 (labeled dataset)
Require: U^0 (Unlabeled dataset)
for $t = 1, 2, \ldots, T$ **do**
  Compute pseudo-labels $\hat{y} = -\,sign\left(f_{\mathcal{A}^t}(x)\right)$ for data points in U^t using algorithm $\mathcal{A}^t$
  Compute p^t as in (6) using loss computed on pseudo-labels.
  Randomly select m data points in U^t according to the probability distribution p^t to form V^t.
  Receive true labels for data points in V^t from the oracle.
  Update S^t: $S^{t+1} \leftarrow S^t \cup V^t$
  Update U^t: $U^{t+1} \leftarrow U^t \backslash V^t$
  Retrain algorithm on V^t to minimize the weighted error $\sum_{x_k \in V^t} \frac{1}{p^t_k} l(x_k, y_k; \mathcal{A}^{t+1})$ to get A^{t+1}.
end for
return Algorithm $\mathcal{A}^{T+1}$

5.1 A Simple Example

To illustrate our algorithm, we present a simple binary classification example here. We generate a 2-class synthetic dataset from a gaussian mixture, and use an online perceptron for training. The loss function is hinge loss. The querying probabilities are computed using (7) and are shown in Figure 1.

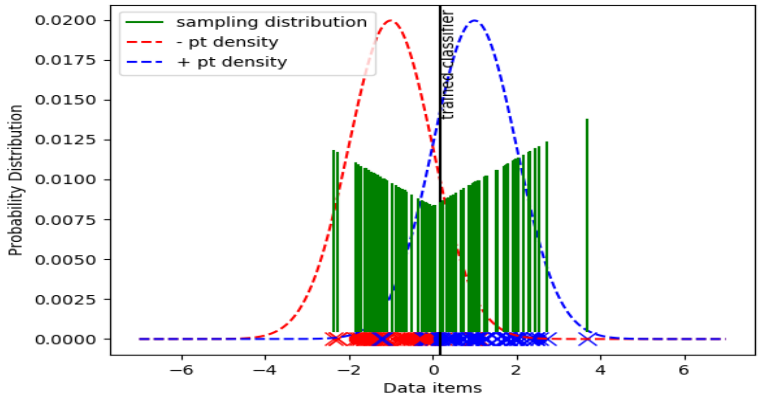

Figure 1: A simple experiment depicting the computation of the optimal querying distribution p^t. The red dashed line is the true distribution for '-' (negative) points, and the blue dashed line is the true distribution for '+' (positive) points. The unlabeled points are marked by x's on the x-axis, while the labeled points are marked by solid circles. The black vertical line is the classifier learnt by the algorithm at this iteration. The lengths of the green vertical lines indicate the optimal sampling probabilities over the unlabeled points.

From Figure 1, we can observe that the sampling probability at an unlabeled point is approximately proportional to the distance of the point from the classification boundary at any iteration. This means that points that are far away from the classification boundary are more likely to be sampled, when compared to the points that are closer to the classification boundary.

6 Analysis

In this section, we derive an upper bound on the true error (i.e. expected loss over the entire domain of the data generating distribution $\mathcal{Z}$) for the Active Learning with Importance Sampling (ALIS) algorithm shown in Algorithm 1.

Theorem 2. *Let* $l^t_{max} = \max_j l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)$. *Define*

$$l^t_{var} = \left(\sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t) \right)^2 - \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2.$$

Using Algorithm 1, the true error at the end of t *iterations can be bounded as follows.*

$$\mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t)] \leq \left| \mathbb{E}_{\mathcal{Z}}[l(x, y; \mathcal{A}^t))] - \frac{1}{n_t} \sum_{j \in [n_t]} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \right| + l^t_{max} \left(m + \frac{1}{3n_t} \left(1 + \sqrt{1 + \frac{18 l^t_{var}}{{l^t_{max}}^2} \log(1/\delta)} \right) \right) \tag{8}$$

Proof. Denote $Z^t_j = l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) - \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)$. Using (4) and the optimal probability distribution given in (6) we get

$$\begin{aligned}
\sum_{j \in [n_t]} Var(Z^t_j) &= \sum_{j \in [n_t]} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)^2 \left(\frac{1}{p^t_{u^t(j)}} - 1 \right) \\
&\leq \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2 \left(\frac{1}{p^t_{u^t(j)}} - 1 \right) \\
&= \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2 \left(\frac{\sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)}{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)} - 1 \right) \\
&= \left(\sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t) \right)^2 - \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)^2 \\
&= l^t_{var}.
\end{aligned}$$

Since the data points for querying are chosen independently (with replacement), we can say that the random variables Q^t_i's are independent. So, Z^t_j's are also independent. Also, $|Z^t_j| \leq l^t_{max}$ for all $j \in [n_t]$, which means they are bounded. Since Z^t_j's are independent bounded random variables with finite variance, we can use Bernstein's inequality to get a high probability bound for (2) as follows.

$$\begin{aligned}
\mathbb{P}\left(\sum_{j \in [n_t]} Z^t_j > \epsilon \right) &< exp\left(-\frac{\epsilon^2/2}{l^t_{var} + \frac{1}{3} l^t_{max} \epsilon} \right) := \delta \\
\Rightarrow \epsilon &= \frac{l^t_{max}}{3} \left(1 + \sqrt{1 + \frac{18 l^t_{var}}{{l^t_{max}}^2} \log(1/\delta)} \right).
\end{aligned}$$

Therefore, with probability at least $1 - \delta$ we have

$$(2) < \frac{l^t_{max}}{3n_t} \left(1 + \sqrt{1 + \frac{18 l^t_{var}}{{l^t_{max}}^2} \log(1/\delta)} \right). \tag{9}$$

Now we can bound (3) as follows.

$$
\begin{aligned}
(3) &= \frac{1}{n_t} \sum_{j \in [n_t]} \frac{Q^t_{u^t(j)}}{p^t_{u^t(j)}} l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t) \\
&= \frac{1}{n_t} \sum_{j \in [n_t]} Q^t_{u^t(j)} \frac{l(x_{u^t(j)}, y_{u^t(j)}; \mathcal{A}^t)}{l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t)} \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t) \\
&\leq \frac{1}{n_t} \sum_{j \in [n_t]} l(x_{u^t(j)}, \hat{y}_{u^t(j)}; \mathcal{A}^t) \sum_{j \in [n_t]} Q^t_{u^t(j)} \\
&\leq m.l^t_{max}. \qquad (10)
\end{aligned}
$$

Combining (9) and (10), we get the desired bound in (8). □

7 Experiments

In this section, we validate the performance of our Active Learning with Importance Sampling (ALIS) algorithm against a Random Sampling (RS) algorithm on a simulated dataset. The querying probability distribution for ALIS algorithm is derived using (6), and for the RS algorithm it is a uniform probability distribution. We use the *online perceptron* algorithm for training over the simulated data.

The data points are 2-dimensional and are drawn $i.i.d.$ from uniform distribution in the range $[-1, 1]^2$. The training set consists of $5,000$ data points, out of which only 5 are labeled initially. The testing set consists of $10,000$ data points. In every iteration we choose 10 data points to be labeled by the oracle. In the first iteration, the perceptron is trained on the 5 initially labeled points. In each subsequent iteration, 10 new points are sampled using the querying probability distribution (ALIS or RS). The labels are obtained for them from the oracle. The perceptron is then retrained only on these 10 newly labeled points.

We ran the experiment 10 times using both the ALIS and RS algorithms for 500 iterations and averaged the test error across the 10 experiments. The results are shown in 2. In the figure, we observe that the average test error for ALIS algorithm converges to zero much faster than the RS algorithm, which indicates a superior generalization performance, as is expected from our theoretical findings.

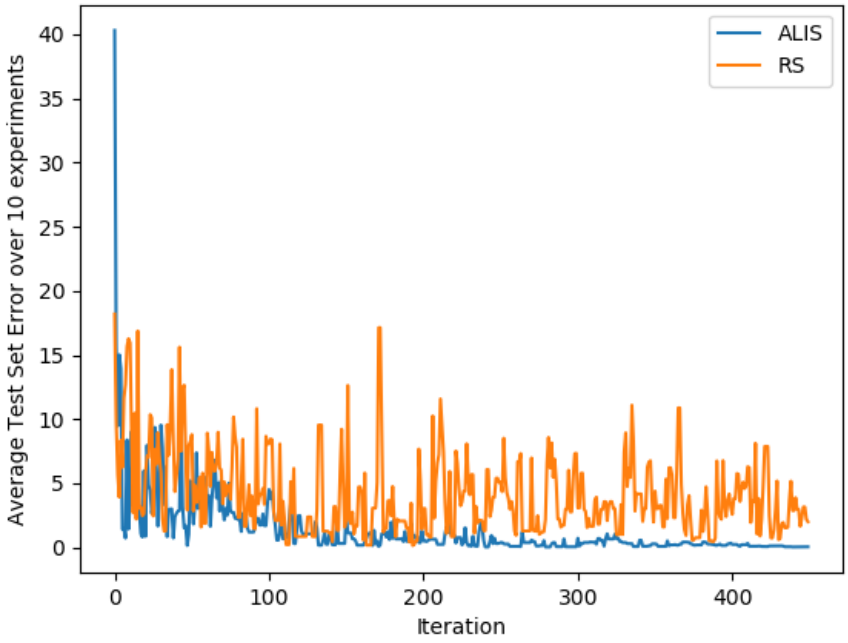

Figure 2: Test error across 450 iterations for ALIS and RS algorithms averaged over 10 experiments.

We also compare ALIS and RS algorithms with a Best Oracle Uncertainty algorithm as given in [Sener and Silvio, 2018]. The Best Oracle Uncertainty algorithm has access to the true labels for the entire training set and uses the true labels to construct the optimal sampling probability distribution, while the ALIS algorithm uses pseudo-labels to do so. Figure 3 compares the test error for ALIS and RS algorithms against the Best Oracle Uncertainty algorithm. From Figure 3 we observe that the performance of ALIS is very close to that of the Best Oracle algorithm. This is suggests that the relaxation of the objective function in (5), replacing the true labels with pseudo-labels, is justified.

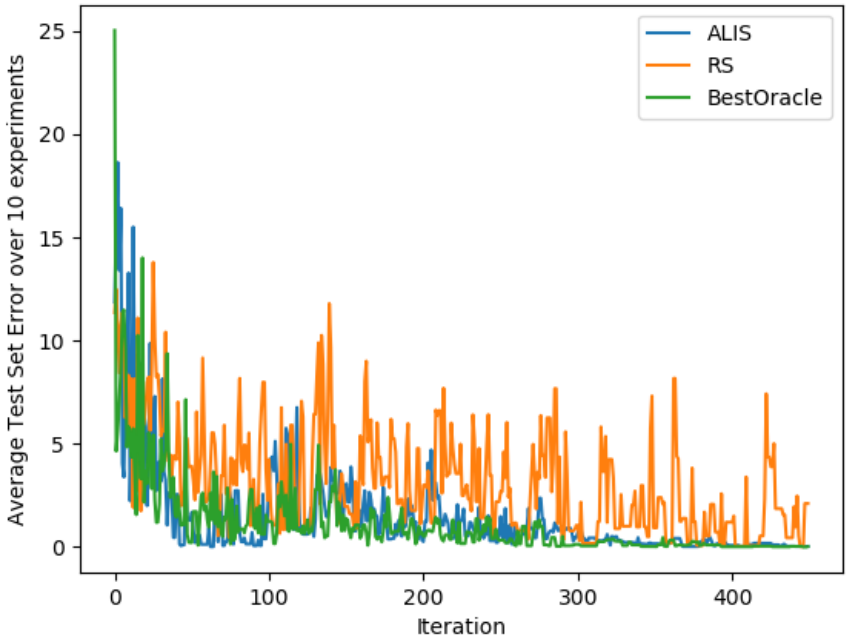

Figure 3: Test error across 450 iterations for ALIS, RS and Best Oracle Uncertainty algorithms averaged over 10 experiments.

8 Conclusion

In this paper, we propose the Active Learning with Importance Sampling (ALIS) algorithm which uses a non-uniform probability distribution to select points to be queried from the oracle. We prove that the probability distribution used by ALIS is optimal for minimizing an upper bound on the true error rate. We then validate our theoretical findings with experiments done on a simulated dataset.

References

Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 49–56. ACM, 2009.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

Michael Prince. Does active learning work? a review of the research. *Journal of engineering education*, 93(3): 223–231, 2004.

Ozan Sener and Savarese Silvio. Active learning for convolutional neural networks: A core-set approach. 2018.

Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Zheng Wang and Jieping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):17, 2015.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *international conference on machine learning*, pages 1–9, 2015.