Online Active Learning of Reject Option Classifiers.

by

kulin Shah, Naresh Manwani

in

AAAI

Report No: IIIT/TR/2020/-1

Centre for Others
International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2020

Online Active Learning of Reject Option Classifiers

Kulin Shah, Naresh Manwani
Machine Learning Lab, KCIS, IIIT Hyderabad, India
kulin.shah@students.iiit.ac.in, naresh.manwani@iiit.ac.in

Abstract

Active learning is an important technique to reduce the number of labeled examples in supervised learning. Active learning for binary classification has been well addressed in machine learning. However, active learning of the reject option classifier remains unaddressed. In this paper, we propose novel algorithms for active learning of reject option classifiers. We develop an active learning algorithm using double ramp loss function. We provide mistake bounds for this algorithm. We also propose a new loss function called double sigmoid loss function for reject option and corresponding active learning algorithm. We offer a convergence guarantee for this algorithm. We provide extensive experimental results to show the effectiveness of the proposed algorithms. The proposed algorithms efficiently reduce the number of label examples required.

1 Introduction

In standard binary classification problems, algorithms return prediction on every example. For any misprediction, the algorithms incur a cost. Many real-life applications involve very high misclassification costs. Thus, for some confusing examples, not predicting anything may be less costly than any misclassification. The choice of not predicting anything for an example is called *reject option* in machine learning literature. Such classifiers are called reject option classifiers.

Reject option classification is very useful in many applications. Consider a doctor diagnosing a patient based on the observed symptoms and preliminary diagnosis. If there is an ambiguity in observations and preliminary diagnosis, the doctor can hold the decision on the treatment. She can recommend to take advanced tests or consult a specialist to avoid the risk of misdiagnosing the patient. The holding response of the doctor is the same as to reject option for the specific patient (da Rocha Neto et al., 2011). On the other hand, the doctor's misprediction can cost huge money for further treatment or the life of a person. In another example, a banker can use the reject option while looking at the loan application of a customer (Rosowsky and Smith, 2013). A banker may choose not to decide based on the information available because of high misclassification cost, and asks for further recommendations or a credit bureau score from the stakeholders. Application of reject option classifiers include healthcare Hanczar and Dougherty (2008); da Rocha Neto et al. (2011), text categorization Fumera, Pillai, and Roli (2003), crowdsourcing Li et al. (2017) etc.

Let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space and $\{+1, -1\}$ be the label space. Examples of the form $(\mathbf{x}, y)$ are generated from an unknown fixed distribution on $\mathcal{X} \times \{+1, -1\}$. A reject option classifier can be described with the help of a function $f : \mathcal{X} \to \mathbb{R}$ and a rejection width parameter $\rho \in \mathbb{R}_+$ as below.

$$h_\rho(f(\mathbf{x})) = 1.\mathbb{I}_{\{f(\mathbf{x})>\rho\}} - 1.\mathbb{I}_{\{f(\mathbf{x})<-\rho\}} - 0.\mathbb{I}_{\{|f(\mathbf{x})|\leq\rho\}} \tag{1}$$

The goal is to learn $f(.)$ and ρ simultaneously. For a given example $(\mathbf{x}, y)$, the performance of reject option classifier $h_\rho(f(.))$ is measured using following loss function.

$$L_d(yf(\mathbf{x}), \rho) = \mathbb{I}_{\{yf(\mathbf{x})\leq-\rho\}} + d\mathbb{I}_{\{|f(\mathbf{x})|\leq\rho\}} \tag{2}$$

where $d \in (0, 0.5)$ is the cost of rejection. A reject option classifier is learnt by minimizing the risk (expectation of loss) under L_d. As L_d is not continuous, optimization of empirical risk under L_d is difficult. Bartlett and Wegkamp (2008); Wegkamp and Yuan (2011) propose a convex surrogate of L_d called generalized hinge loss. They learn the reject option classifier using risk minimization algorithms based on generalized hinge loss. Grandvalet et al. (2008) propose another convex surrogate of L_d called double hinge loss and corresponding risk minimization approach for reject option classification. Manwani et al. (2015); Shah and Manwani (2019) propose double ramp loss based approaches for reject option classification. Double ramp loss is a non-convex bounded loss function. All these approaches assume that we have plenty of labeled data available.

In general, classifiers learned with a large amount of training data can give better generalization on testing data. However, in many real-life applications, it can be costly and difficult to get a large amount of labeled data. Thus, in many cases, it is desirable to ask the labels of the examples selectively. This motivates the idea of active learning. Active learning selects more informative examples and queries labels of those examples. Active learning of standard binary classifiers has

been well-studied (Dasgupta, Kalai, and Monteleoni, 2009; Bachrach, Fine, and Shamir, 1999; Tong and Koller, 2002). In El-Yaniv and Wiener (2012), authors reduce active learning for the usual binary classification problem to learning a reject option classifier to achieve faster convergence rates. However, active learning of reject option classifiers has remained an unaddressed problem. In this paper, we propose online active learning algorithms to reject option classification.

Let us reconsider the example where the banker uses the reject option classifier for selecting the loan applications. Consider a loan application that satisfies the basic requirements. Thus, the banker is not clear about using the hold option. On the other hand, she is also not sure enough to approve the application. Such cases are instrumental in defining the separation rule between accepting the loan application and holding it for further investigation. This motivates us to think that one can use active learning to ask the labels of selective examples as described above while learning the reject option classifier.

A broad class of active learning algorithms is inspired by the concept of a margin between the two categories. Thus, an example, which falls in the margin area of the current classifier, carries more information about the decision boundary. On the other hand, examples which are correctly classified with good margin or misclassified by a good margin, give less knowledge of the decision boundary. Margin examples can bring more changes to the existing classifier. Thus, querying the label of margin examples is more desirable than the other two kinds of examples.

A reject option classifier can be viewed as two parallel surfaces with the rejection area in between. Thus, active learning of the reject option classifier becomes active learning of two surfaces in parallel with a shared objective. This shared objective is nothing but to minimize the sum of L_d losses over a sequence of examples. In Manwani et al. (2015), the authors propose a risk minimization approach based on double ramp loss (L_{dr}) for learning the reject option classifier. In Manwani et al. (2015), it is shown that at the optimality, the two surfaces can be represented using only those examples which are close to them. Examples that are far from the two surfaces do not participate in the representation of the surfaces. This motivates us to use double ramp loss for developing an active learning approach to reject option classifiers.

Our Contributions

We make the following contributions in this paper.

- We propose an active learning algorithm based on double ramp loss L_{dr} to learn a linear and non-linear classifier. We give bounds to the number of rejected examples and misclassification rates for un-rejected examples.
- We propose a smooth non-convex loss called double sigmoid loss (L_{ds}) for reject option classification.
- We propose an active learning algorithm based on L_{ds} to learn both linear and non-linear classifiers. We also give convergence guarantees for the proposed algorithm.
- We present extensive simulation results for both proposed active learning algorithms for linear as well as non-linear classification boundaries.

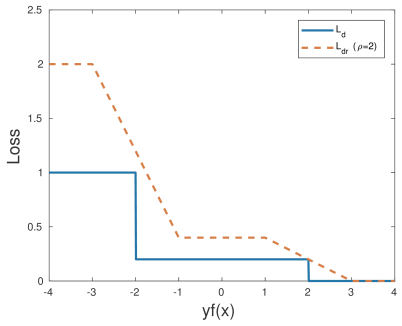

Figure 1: Double Ramp Loss with $\rho = 2$

2 Proposed Approach: Active Learning Inspired by Double Ramp Loss

Active learning algorithm does not ask the label in every trial. We denote the instance presented to algorithm at trial t by $\mathbf{x}_t$. Each $\mathbf{x}_t \in \mathcal{X}$ is associated with a unique label $y_t \in \{-1, 1\}$. The algorithm calculates $f_t(\mathbf{x}_t)$ and outputs the decision using eq.(1). Based on $f_t(\mathbf{x}_t)$, the active learning algorithm decides whether to ask label or not. Guillory, Chastain, and Bilmes (2009) shows that online active learning algorithms can be viewed as stochastic gradient descent on non-convex loss function therefore, we use a non-convex loss function *Double ramp loss* L_{dr} (Manwani et al., 2015) to derive our first active learning approach. L_{dr} is defined as follows.

$$L_{dr}(yf(\mathbf{x}), \rho) = d\Big[\big[1 - yf(\mathbf{x}) + \rho\big]_+ - \big[-1 - yf(\mathbf{x}) + \rho\big]_+\Big] + (1-d)\Big[\big[1 - yf(\mathbf{x}) - \rho\big]_+ - \big[-1 - yf(\mathbf{x}) - \rho\big]_+\Big]$$

Here $[a]_+ = \max(0, a)$ and d is the cost of rejection. Figure 1 shows the plot of double ramp loss for $\rho = 2$.

We first consider developing active learning algorithm for linear classifiers (i.e. $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$). We use stochastic gradient descent (SGD) to derive double ramp loss based active learning algorithm. Parameters update equations using SGD are as follows.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} L_{dr}(y_t f(\mathbf{x}_t), \rho_t) = \begin{cases} \mathbf{w}_t + \eta d y_t \mathbf{x}_t, & \rho_t - 1 \leq y_t f(\mathbf{x}_t) \leq \rho_t + 1 \\ \mathbf{w}_t + \eta(1-d) y_t \mathbf{x}_t & -\rho_t - 1 \leq y_t f(\mathbf{x}_t) \leq -\rho_t + 1 \\ \mathbf{w}_t & \text{otherwise} \end{cases}$$

$$\rho_{t+1} = \rho_t - \eta \nabla_{\rho_t} L_{dr}(y_t f(\mathbf{x}_t), \rho_t) = \begin{cases} \rho_t - \eta d, & \rho_t - 1 \leq y_t f(\mathbf{x}_t) \leq \rho_t + 1 \\ \rho_t + \eta(1-d), & -\rho_t - 1 \leq y_t f(\mathbf{x}_t) \leq -\rho_t + 1 \\ \rho_t & \text{otherwise} \end{cases}$$

Where η is the step-size. We see that the parameters are updated only when $|f_t(\mathbf{x}_t)| \in [\rho_t - 1, \rho_t + 1]$. For the rest of the regions, the gradient of the loss L_{dr} is zero therefore, there won't be any update when an example $\mathbf{x}_t$ is such that $|f_t(\mathbf{x}_t)| \notin [\rho_t - 1, \rho_t + 1]$. Thus, there is no need to query the label when $|f_t(\mathbf{x}_t)| \notin [\rho_t - 1, \rho_t + 1]$. We only query the labels when $|f_t(\mathbf{x}_t)| \in [\rho_t - 1, \rho_t + 1]$. Thus, we ask the

label of the current example only if it falls in the linear region of the loss L_{dr}. This is the same way any margin based active learning approach updates the parameters. If the algorithm does not query the label y_t, the parameters $(\mathbf{w}, \rho)$ are not updated. Thus, we define the query function Q_t as follows.

$$Q_t = \begin{cases} 1 & \text{if } \rho_t - 1 \leq |f(\mathbf{x}_t)| \leq \rho_t + 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The detailed algorithm is given in Algorithm 1. We call it DRAL (double ramp loss based active learning). DRAL can be easily extended for learning nonlinear classifiers using kernel trick and is described in the supplementary file.

Algorithm 1 **D**ouble **R**amp Loss **A**ctive **L**earning (DRAL)

Input: $d \in (0, 0.5)$, step size η
Output: Weight vector $\mathbf{w}$, Rejection width ρ
Initialize: $\mathbf{w}_1 = \mathbf{0}, \rho_1 = 1$
for $t = 1, \ldots, T$ **do**
  Sample $\mathbf{x}_t \in S$
  Set $f_t(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$
  if $\rho_t - 1 \leq |f_t(\mathbf{x}_t)| \leq \rho_t + 1$ **then**
    Set $Q_t = 1$
    Query the label y_t of $\mathbf{x}_t$.
    if $(\rho_t - 1 \leq \ y_t f_t(\mathbf{x}_t) \ \leq \rho_t + 1)$ **then**
      $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta d y_t \mathbf{x}_t$.
      $\rho_{t+1} = \rho_t - \eta d$
    else if $(-\rho_t - 1 \leq y_t f_t(\mathbf{x}_t) \leq -\rho_t + 1)$ **then**
      $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta(1-d) y_t \mathbf{x}_t$
      $\rho_{t+1} = \rho_t + \eta(1-d)$
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t$
    $\rho_{t+1} = \rho_t$

Mistake Bounds for DRAL

In this section, we derive the mistake bounds of DRAL. Before presenting the mistake bounds, we begin by presenting a lemma which would facilitate the following mistake bound proofs. Let $f_t(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$. We define the following.[1]

$$\begin{cases} C_t = \mathbb{I}_{\{\rho_t \leq y_t f_t(\mathbf{x}_t) \leq \rho_t + 1\}} & R_{1t} = \mathbb{I}_{\{\rho_t - 1 \leq y_t f_t(\mathbf{x}_t) \leq \rho_t\}} \\ R_{2t} = \mathbb{I}_{\{-\rho_t \leq y_t f_t(\mathbf{x}_t) \leq -\rho_t + 1\}} & M_t = \mathbb{I}_{\{-\rho_t - 1 \leq y_t f_t(\mathbf{x}_t) \leq -\rho_t\}} \end{cases} \tag{4}$$

Lemma 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \{-1, 1\}$ for all $t \in [T]$.[2] Given C_t, R_{1t}, R_{2t} and M_t as defined in eq.(4) and $\alpha > 0$, the following bound holds for any $\mathbf{w}$.*

$$\alpha^2 \|\mathbf{w}\|^2 + (1 - \alpha\rho)^2 + 2\alpha\eta \sum_{t=1}^{T} L_{dr}(y_t f(\mathbf{x}_t), \rho) \geq$$
$$\sum_{t=1}^{T} [\, C_t + R_{1t} \,] \big[2\alpha\eta d + 2\eta (L_{dr}(y_t f_t(\mathbf{x}_t), \rho_t) - d)$$
$$- \eta^2 d^2 (\|\mathbf{x}_t\|^2 + 1) \big] + \sum_{t=1}^{T} [\, R_{2t} + M_t \,] \big[2\alpha\eta (1 + d)$$
$$+ 2\eta (L_{dr}(y_t f_t(\mathbf{x}_t), \rho_t) - d - 1) - \eta^2 (1-d)^2 (\|\mathbf{x}_t\|^2 + 1) \big]$$

where $f(\mathbf{x}_t) = \mathbf{w} \cdot \mathbf{x}_t$ and $f_t(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$.

The proof is given in the Supplementary file. Now, we will find the bounds on rejection rate and mis-classification rate.

Theorem 2. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \{-1, 1\}$ and $\|\mathbf{x}_t\| \leq R$ for all $t \in [T]$. Assume that there exists a $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ and ρ such that $L_{dr}(y_t f(\mathbf{x}_t), \rho) = 0$ for all $t \in [T]$.*

1. *Number of examples rejected by DRAL (Algorithm 1) among those for which the label was asked in this sequence is upper bounded as follows.*

$$\sum_{t: Q_t = 1} [R_{1t} + R_{2t}] \leq \alpha^2 \|\mathbf{w}\|^2 + (1 - \alpha\rho)^2$$

where $\alpha = \max\left(\frac{1 + \eta^2 d^2 (R^2+1) + 2\eta d}{2\eta d}, \frac{1 + \eta^2 (1-d)^2 (R^2+1) + 2\eta(1-d)}{2\eta(1+d)} \right)$.

2. *Number of examples mis-classified by DRAL (Algorithm 1) among those for which the label was asked in this sequence is upper bounded as follows.*

$$\sum_{t: Q_t = 1} M_t \leq \alpha^2 \|\mathbf{w}\|^2 + (1 - \alpha\rho)^2$$

where $\alpha = \max\left(\frac{\eta d (R^2+1) + 2}{2}, \frac{1 + \eta^2 (1-d)^2 (R^2+1) + 2\eta(1-d)}{2\eta(1+d)} \right)$.

The proof is given in the Supplementary file. The above theorem assumes that there exists $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ and ρ such that $L_{dr}(y_t f(\mathbf{x}_t), \rho) = 0$ for all $t \in [T]$. This means that the data is linearly separable. In such a case, the number of mistakes made by the algorithm on unrejected examples as well as the number of rejected examples are upper bounded by a complexity term and are independent of T. Now, we derive the bounds when the assumption $L_{dr}(y_t f(\mathbf{x}_t), \rho) = 0, \ t \in [T]$ does not hold for any $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ and ρ.

Theorem 3. *Let $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathcal{X}$ and $y_t \in \{-1, 1\}$ and $\|\mathbf{x}_t\| \leq R$ for all $t \in [T]$. Then, for any given $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ and ρ, we observe the following.*

1. *Number of rejected examples by DRAL (Algorithm 1) among those for which the label was asked in this sequence is upper bounded as follows.*

$$\sum_{t: Q_t = 1} [R_{1t} + R_{2t}] \leq \alpha^2 \|\mathbf{w}\|^2 + (1 - \alpha\rho)^2 + \sum_{t=1}^{T} 2\eta\alpha L_{dr}(y_t f(\mathbf{x}_t), \rho)$$

where $\alpha = \max \begin{cases} \frac{1 + \eta^2 d^2 (R^2+1) + 2\eta d}{2\eta d} \\ \frac{1 + \eta^2 (1-d)^2 (R^2+1) + 2\eta(1-d)}{2\eta(1+d)} \end{cases}$.

[1] $\mathbb{I}_{\{A\}}$ takes value 1 when A is true and 0 otherwise.
[2] Here, $[T]$ denotes the sequence $1, \ldots, T$.

2. *The number of misclassified examples by DRAL (Algorithm 1) is upper bounded as follows.*

$$\sum_{t:Q_t=1} M_t \leq \alpha^2\|\mathbf{w}\|^2 + (1-\alpha\rho)^2 + \sum_{t=1}^{T} 2\eta\alpha L_{dr}(y_t f(\mathbf{x}_t), \rho)$$

$$\textit{where } \alpha = \max\left\{\frac{\eta d(R^2+1)+2}{2}, \frac{1+\eta^2(1-d)^2(R^2+1)+2\eta(1-d)}{2\eta(1+d)}\right\}.$$

The proof is given in the Supplementary file. We see that when the data is not linearly separable, the number of mistakes made by the algorithm is upper bounded by the sum of complexity term and sum of the losses using a fixed classifier.

3 Active Learning Using Double Sigmoid Loss Function

We observe that double ramp loss is not smooth. Moreover, L_{dr} is constant whenever $yf(\mathbf{x}) \in [\rho+1, \infty) \cup (-\infty, -\rho-1] \cup [-\rho+1, \rho-1]$. Thus, when loss L_{dr} for an example $\mathbf{x}$ falls in any of these three regions, the gradient of the loss becomes zero. The zero gradient causes no update. Thus, there is no benefit of asking the labels when an example falls in one of these regions. However, we don't want to ignore these regions completely. To capture the information in these regions, we need to change the loss function in such a way that the gradient does not vanish completely in these regions. To ensure that, we propose a new loss function.

Double Sigmoid Loss

We propose a new loss function for reject option classification by combining two sigmoids as follows. We call it *double sigmoid loss* function L_{ds}.

$$L_{ds}(yf(\mathbf{x}), \rho) = 2d\sigma(yf(\mathbf{x}) - \rho) + 2(1-d)\sigma(yf(\mathbf{x}) + \rho)$$

where $\sigma(a) = (1+e^{\gamma a})^{-1}$ is the sigmoid function ($\gamma > 0$). Figure 2 shows the double sigmoid loss function. L_{ds} is a smooth non-convex surrogate of loss L_d (see eq.(2)). We also see that for the double sigmoid loss, the gradient in the regions $yf(\mathbf{x}) \in [\rho+1, \infty) \cup (-\infty, -\rho-1] \cup [-\rho+1, \rho-1]$ does not vanish unlike double ranp loss. Below we establish

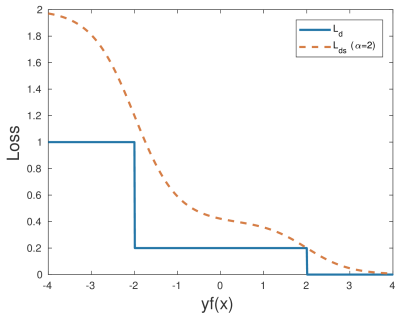

Figure 2: Double sigmoid loss with $\gamma = 2$.

that the loss L_{ds} is β-smooth.[3]

Lemma 4. *Assuming* $\|\mathbf{x}\| \leq \mathbf{R}$, *Double sigmoid loss* $L_{ds}(yf(\mathbf{x}), \rho)$ *is* $\beta-$*smooth with constant* $\beta = \frac{\gamma^2}{5}\left[\mathbf{R}^2+1\right]$.

The proof is given in the supplementary file.

Query Probability Function

In the case of DRAL, we saw that the gradient of L_{dr} becomes nonzero only in the region $yf(\mathbf{x}) \in [\rho-1, \rho+1]$. So, we ask the labels only when examples fall in this region. However, in case of double sigmoid loss, the gradient does not vanish. Thus, to perform active learning using L_{ds}, we need to ask the labels selectively.

We propose a query probability function to set the label query probability at trial t. The query probability function should carry the following properties. In the loss L_d (see eq.(2)), we see two transitions. One at $yf(\mathbf{x}) = \rho$ (transition between correct classification and rejection) and another at $yf(\mathbf{x}) = -\rho$ (transition between rejection and misclassification). Any example falling closer to one of these transitions captures more information about the two transitions. We want the query probability function to be such that it gives higher probabilities near these transitions. Examples that are correctly classified with a good margin, examples misclassified with a considerable margin, and examples in the middle of the reject region do not carry much information. Such examples are also situated away from the transition regions. Thus, query probability should decrease as we move away from these decision boundaries. Therefore, we ask the label in these regions with less probability. Considering these desirable properties, we propose the following query probability function.

$$p_t = 4\,\sigma(|f_t(\mathbf{x}_t)| - \rho_t)\,(1 - \sigma(|f_t(\mathbf{x}_t)| - \rho_t)) \qquad (5)$$

where $f_t(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$. Figure 3 shows the graph of the query probability function. We see that the probability function has two peaks. One peak is at $yf(\mathbf{x}) = \rho$ (transition between correct classification and rejection) and another at $yf(\mathbf{x}) = -\rho$ (transition between rejection and misclassification).

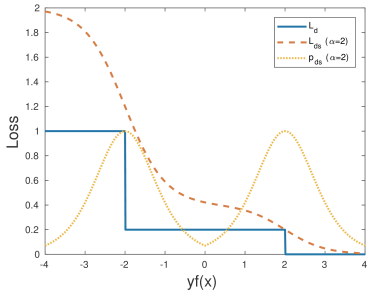

Figure 3: Query Probability Function

[3] A function f is β-smooth if for all $x, y \in$ Domain(f),

$$\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|.$$

Double Sigmoid Based Parameter Updates

The parameter update equations using L_{ds} is as follows.

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{w}_t - \eta \nabla_{\mathbf{w}_t} L_{ds}(y_t f(\mathbf{x}_t), \rho_t) \\ &= \mathbf{w}_t - 2y_t \alpha \mathbf{x}_t \Big[d\sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\right) \\ &+ (1-d)\sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\right) \Big] \quad (6) \\ \rho_{t+1} &= \rho_t - \eta \nabla_{\rho_t} L_{ds}(y_t f(\mathbf{x}_t), \rho_t) \\ &= \rho_t + 2\alpha \Big[d\sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\right) \\ &- (1-d)\sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\right) \Big] \quad (7)\end{aligned}$$

Now, we will explain the update equations for $\mathbf{w}$ and ρ.

1. When an example is correctly classified with good margin (i.e. $y_t f_t(\mathbf{x}_t) >> 0$) then the active learning algorithm will update $\mathbf{w}$ by a small factor of $y_t \mathbf{x}_t$ and will reduce the rejection width (ρ) because for $y_t f_t(\mathbf{x}_t) >> 0$, $d\sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\right) > (1-d)\sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\right)$.
2. When an example is misclassified with good margin (i.e. $y_t f_t(\mathbf{x}_t) << 0$) then the active learning algorithm will update $\mathbf{w}$ by a large factor of $y_t \mathbf{x}_t$ and will increase the rejection width (ρ) because for $y_t f_t(\mathbf{x}_t) << 0$, $d\sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) - \rho_t)\right) < (1-d)\sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\left(1 - \sigma(y_t f_t(\mathbf{x}_t) + \rho_t)\right)$.

We use the acronym DSAL for double sigmoid based active learning. DSAL is described in Algorithm 2.

Algorithm 2 **D**ouble **S**igmoid Loss **A**ctive **L**earning (DSAL)

Input: $d \in (0, 0.5)$, step size η
Output: Weight vector $\mathbf{w}$, Rejection width ρ.
Initialize: $\mathbf{w}_1, \rho_1$
for $t = 1, .., T$ **do**
  Sample $\mathbf{x}_t \in \mathbb{R}^d$
  Set $f_t(\mathbf{x}_t) = \mathbf{w}_t \cdot \mathbf{x}_t$
  Set $p_t = 4\sigma(|f_t(\mathbf{x}_t)| - \rho_t)\left(1 - \sigma(|f_t(\mathbf{x}_t)| - \rho_t)\right)$
  Randomly sample $z_t \in \{0, 1\}$ from Bernoulli(p_t).
  if $z_t == 1$ **then**
    Query the label y_t of $\mathbf{x}_t$.
    Find $\mathbf{w}_{t+1}$ using eq.(6).
    Find ρ_{t+1} using eq.(7).
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t$.
    $\rho_{t+1} = \rho_t$.

Convergence of DSAL

In the case of DRAL, the mistake bound analysis was possible as L_{dr} increases linearly in the regions where its gradient is nonzero. However, we don't see similar behavior in double sigmoid loss L_{ds}. Thus, we are not able to carry out the same analysis here. Instead, we here show the convergence of DSAL to local minima. For which, we borrow the techniques from online non-convex optimization. In online non-convex optimization, it is challenging to converge towards a global minimizer. It is a common practice to state the convergence guarantee of an online non-convex optimization algorithm by showing it's convergence towards an ϵ-approximate stationary point. In our case, it means that for some t, $\|\nabla L_{ds}(y_t f_t(\mathbf{x}_t), \rho_t)\|^2 \leq \epsilon$. To prove the convergence of DSAL, we use the notion of local regret defined in (Hazan, Singh, and Zhang, 2017) .

Definition 5. *The local regret for an online algorithm is*

$$\mathcal{R}(T) = \sum_{t=1}^{T} \|\nabla L_{ds}(y_t f_t(\mathbf{x}_t), \rho_t)\|^2.$$

where T is the total number of trials. (Defined in (Hazan, Singh, and Zhang, 2017))

Thus, in each trial, we incur a regret, which is the squared norm of the gradient of the loss. When we reach a stationary point, the gradient will vanish and hence the norm. Note that the convergence here requires that the objective function should be β-smooth. In this case, L_{ds} holds that property, as shown in Lemma 4. Thus, we can use the convergence approach proposed in (Hazan, Singh, and Zhang, 2017).[4]

Theorem 6. *If we choose $\eta = \frac{5}{\gamma^2\left[\mathbf{R}^2+1\right]}$, then using smoothness condition of $L_{ds}(yf(\mathbf{x}), \rho)$, the local regret of DSAL algorithm is bounded as follows.*

$$\mathcal{R}(T) \leq \frac{4\gamma^2}{5}\left(\mathbf{R}^2 + 1\right)(T+1)$$

The proof is given in the supplementary file. To prove that DSAL reaches $\epsilon-$stationary point in expectation over iterates, we use following result of (Hazan, Singh, and Zhang, 2017).

$$\mathop{\mathbb{E}}_{t \sim \text{Unif}[T]}\left[\|\nabla L_{ds}(yf(\mathbf{x}), \rho)\|^2\right] \leq \frac{\mathcal{R}(T)}{T} \quad (8)$$

Corollary 7. *For DSAL algorithm,*

$$\mathop{\mathbb{E}}_{t \sim \textit{Unif}[T]}\left[\|\nabla L_{ds}(yf(\mathbf{x}), \rho)\|^2\right] \leq \frac{4\gamma^2}{5}\left(\mathbf{R}^2 + 1\right)\left(1 + \frac{1}{T}\right) \quad (9)$$

Using theorem 6 and eq. (8), we can get the required result of the Corollary. In the Corollary, We see that upper bound on the expectation of the square of the gradient is inversely proportional to T; hence, decreases as the total number of trials T increases. It means that the probability of DSAL algorithm reaches to $\epsilon-$stationary point increases as T increases.

4 Experiments

We show the effectiveness of the proposed active learning approaches on Gisette, Phishing and Guide datasets available on UCI ML repository (Lichman, 2013).

[4] L_{dr} does not have sufficient smoothness properties required in (Hazan, Singh, and Zhang, 2017). Thus, we do not present these convergence results for DRAL.

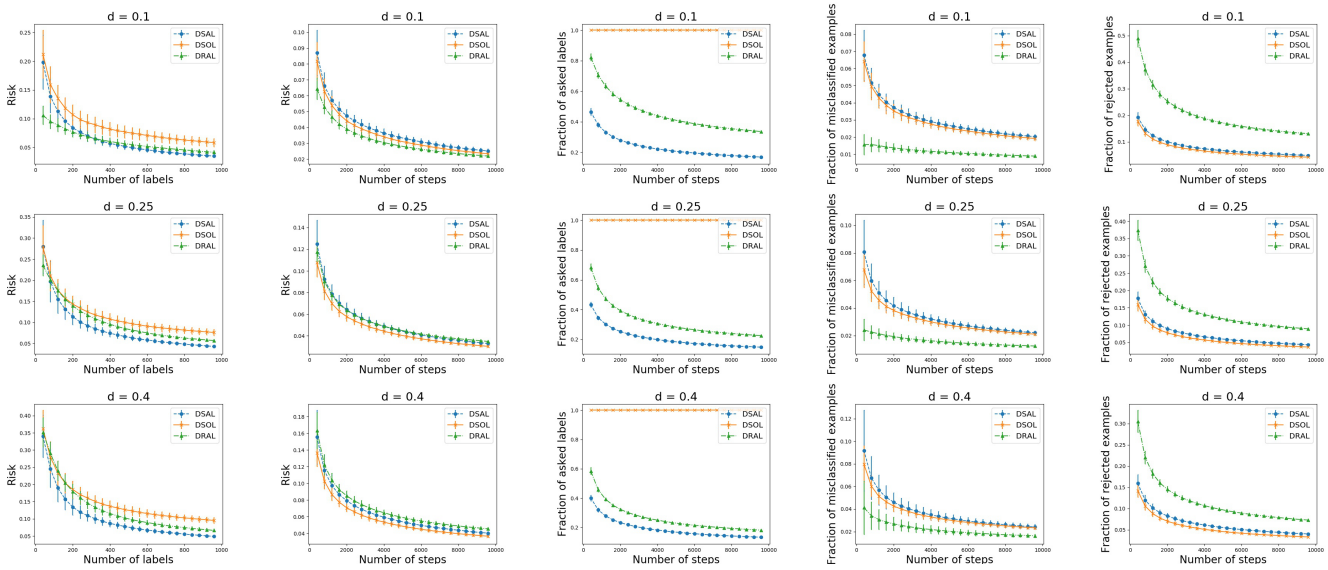

Figure 4: Comparison plots for Gisette dataset with linear Kernel function.

Experimental Setup

We evaluate the performance of our approaches to learning linear classifiers. In all our simulations, we initialize step size by a small value, and after every trial, step size decreases by a small constant. Parameter α in the double sigmoid loss function is chosen to minimize the average risk and average fraction of queried labels (averaged over 100 runs).

We need to show that the proposed active learning algorithms are effectively reducing the number of labeled examples required while achieving the same accuracy as online learning. Thus, we compare the active learning approaches with an online algorithm that updates the parameters using gradient descent on the double sigmoid loss at every trial. We call this online algorithm as DSOL (double sigmoid loss based online learning).

Simulation Results

We report the results for three different values of $d \in \{0.1, 0.25, 0.4\}$. The results provided here are based on 100 repetitions of a total number of trial (T) equal to 10000. For every value of d, we find the average of risk, the fraction of asked labels, fraction of misclassified examples, and fraction of rejected examples over 100 repetitions. We plotted the average of each quantity (e.g., risk, the fraction of asked labels, etc.) as a function of $t \in [T]$. Moreover, the standard deviation of the quantity is denoted by error bar in figures. Figure 4, 5 and 6 show experimental results for Gisette and Phishing and Guide datasets. We observe the following.

- **Label Complexity Versus Risk:** The first column in each figure shows how the risk goes down with the number of asked labels. For Gisette and Phishing datasets, given the number of queried labels, both DSAL and DRAL achieve lower risk compared to DSOL. For Guide dataset, DSAL always makes lower risk compared to DSOL for a given number of queried labels. For Gisette and Guide datasets, DSAL achieves lower risk compared to DRAL with the same number of label queries. For Phishing dataset, DSAL and DRAL perform comparably.

- **Average Risk:** The second column in all the figures shows how the average risk (average of L_d) goes down with the number of steps (t). In all the cases, we see that the risk increases with increasing the value of d. We understand that the average risk of DSAL is higher than DRAL for Gisette and Phishing datasets and all values of d. For Guide dataset, DSAL always achieves lower risk compared to DRAL.

  For Gisette and Guide datasets, DSAL achieves similar risk as DSOL. For Phishing dataset, DSOL performs marginally better than DSAL and DRAL. DRAL does better risk minimization compared to DSOL for Phishing dataset. For Guide dataset, DRAL performs comparable to DSOL as t becomes larger except for $d = 0.1$.

- **Average Fraction of Asked Labels:** Third column in all the figures show the fraction of labels asked for a given time step t. We observe that the fraction of asked labels decreases with increasing d. For Gisette and Phishing datasets, DSAL asks significantly less number of labels as DRAL. This happens because DRAL asks labels every time in a specific region and completely ignores other regions, but DSAL asks labels in every region with some probability. For Guide dataset, the fraction of labels asked to become the same for both DSAL and DRAL as t becomes larger.

- **Average Fraction of Misclassified Examples:** The fourth column of all the figures, shows how the average fraction of misclassified examples goes down with t. We observe that the misclassification rate goes up with increasing d. We see that DRAL achieves a minimum average

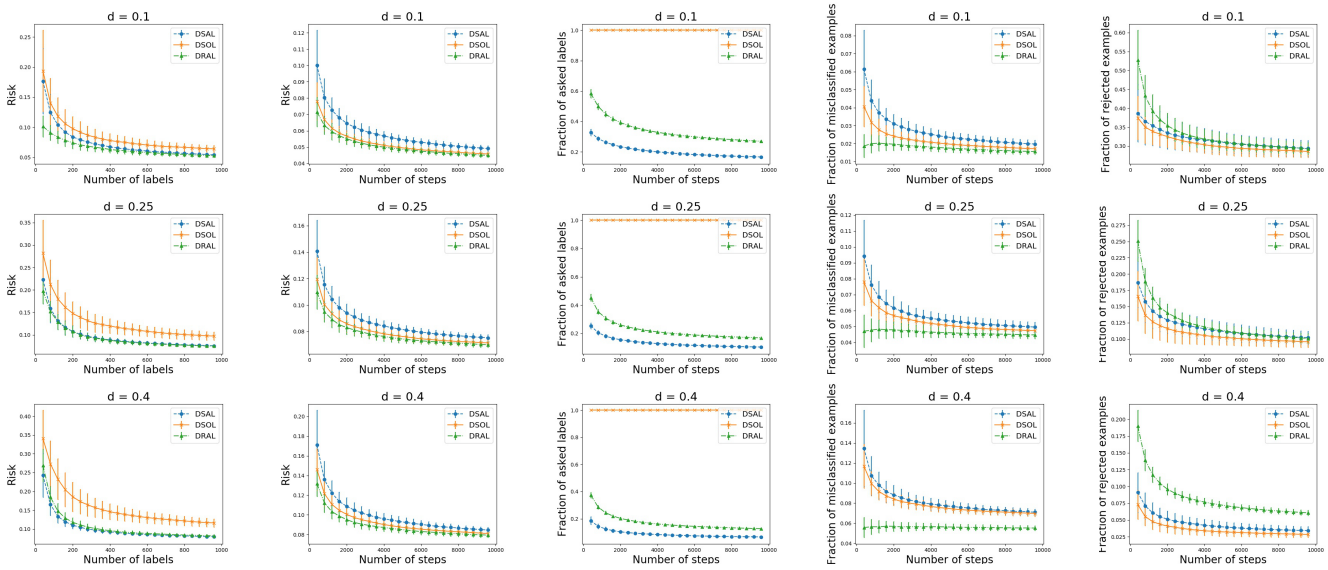

Figure 5: Comparison plots for Phishing dataset with linear Kernel function.

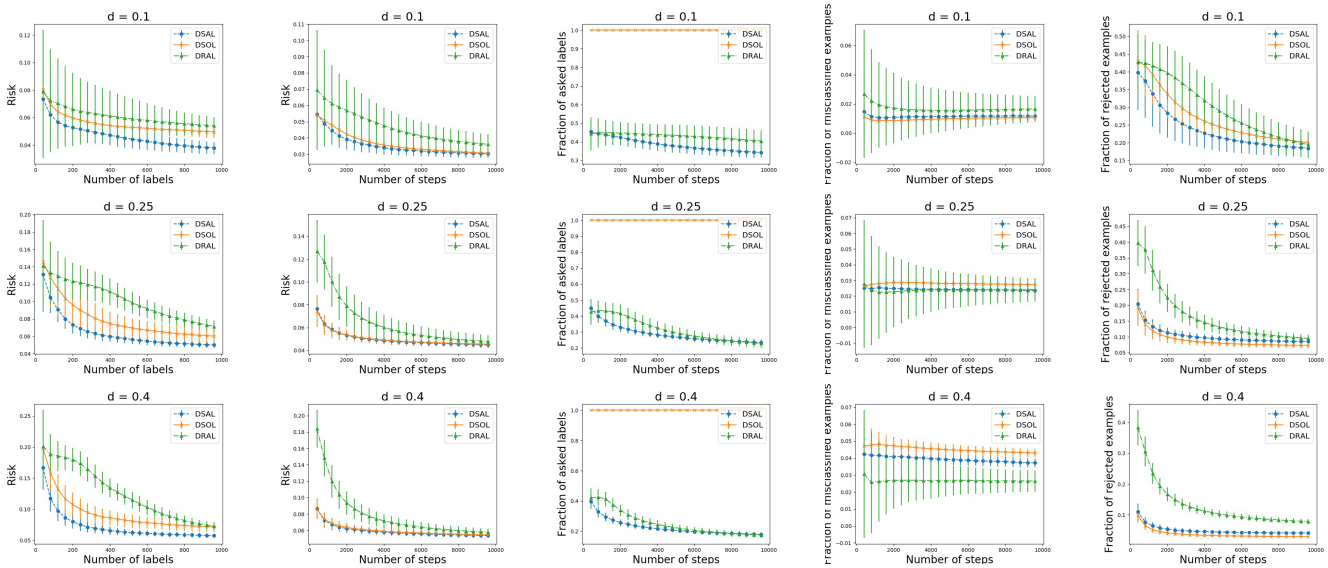

Figure 6: Comparison plots for Guide dataset with polynomial kernel function.

misclassification rate in all the cases compared to DSOL and DSAL except for the Guide dataset with $d = 0.1$ value. For Gisette and Phishing datasets, DSAL achieves a comparable average misclassification rate compared to DSOL for all the cases. For Guide dataset, DSAL achieves a lower misclassification rate compared to DSOL except for $d = 0.1$.

- **Average Fraction of Rejected Examples:** The fifth column in each figure shows how the rejection rate goes down with steps t. We see that the average fraction of rejected examples is higher in DRAL than DSAL and DSOL. Also, the rejection rate decreases with increasing d.

Thus, we see that the proposed active learning algorithms DRAL and DSAL effective reduce the number of labels required for learning the reject option classifier and perform better compared to online learning.

5 Conclusion

In this paper, we have proposed novel active learning algorithms DRAL and DSAL. We presented mistake bounds for DRAL and convergence results for DSAL. We experimentally show that the proposed active learning algorithms reduce the number of labels required while maintaining a similar performance as online learning.

References

Bachrach, R.; Fine, S.; and Shamir, E. 1999. Query by committee, linear separation and random walks. In *Proceedings of the 4th European Conference on Computational Learning Theory*, EuroCOLT '99, 34–49.

Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research.* 9:1823–1840.

da Rocha Neto, A. R.; Sousa, R.; de A. Barreto, G.; and Cardoso, J. S. 2011. Diagnostic of pathology on the vertebral column with embedded reject option. In *Pattern Recognition and Image Analysis*, 588–595.

Dasgupta, S.; Kalai, A. T.; and Monteleoni, C. 2009. Analysis of perceptron-based active learning. *J. Mach. Learn. Res.* 10:281–299.

El-Yaniv, R., and Wiener, Y. 2012. Active learning via perfect selective classification. *J. Mach. Learn. Res.* 13(1):255–279.

Fumera, G.; Pillai, I.; and Roli, F. 2003. Classification with reject option in text categorisation systems. In *12th International Conference on Image Analysis and Processing, 2003.Proceedings.*, 582–587.

Grandvalet, Y.; Rakotomamonjy, A.; Keshet, J.; and Canu, S. 2008. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems (NIPS)*, 537–544.

Guillory, A.; Chastain, E.; and Bilmes, J. 2009. Active learning as non-convex optimization. In van Dyk, D., and Welling, M., eds., *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, 201–208. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR.

Hanczar, B., and Dougherty, E. R. 2008. Classification with reject option in gene expression data. *Bioinformatics* 24(17):1889–1895.

Hazan, E.; Singh, K.; and Zhang, C. 2017. Efficient regret minimization in non-convex games. *CoRR* abs/1708.00075.

Li, Q.; Vempaty, A.; Varshney, L.; and Varshney, P. 2017. Multi-object classification via crowdsourcing with a reject option. *IEEE Transactions on Signal Processing* 65(4):1068–1081.

Lichman, M. 2013. UCI machine learning repository.

Manwani, N.; Desai, K.; Sasidharan, S.; and Sundararajan, R. 2015. Double ramp loss based reject option classifier. In *19th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, 151–163.

Rosowsky, Y. I., and Smith, R. E. 2013. Rejection based support vector machines for financial time series forecasting. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 1161–1167.

Shah, K., and Manwani, N. 2019. Sparse reject option classifier using successive linear programming. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*.

Tong, S., and Koller, D. 2002. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.* 2:45–66.

Wegkamp, M., and Yuan, M. 2011. Support vector machines with a reject option. *Bernoulli* 17(4):1368–1385.