

# Double Ramp Loss Based Reject Option Classifier

Naresh Manwani<sup>1</sup>, Kalpit Desai<sup>2</sup>, Sanand Sasidharan<sup>1</sup>, and Ramasubramanian Sundararajan<sup>3</sup>

<sup>1</sup> Data Mining Lab, GE Global Research, JFWTC, Whitefield, Bangalore-560066, (Naresh.Manwani@ge.com, Sanand.Sasidharan@ge.com),

<sup>2</sup> Bidgely, Bangalore (kvdesai@gmail.com),

<sup>3</sup> Sabre Airline Solutions, Bangalore (gs.ramsu@gmail.com)

**Abstract.** We consider the problem of learning reject option classifiers. The goodness of a reject option classifier is quantified using  $0 - d - 1$  loss function wherein a loss  $d \in (0, .5)$  is assigned for rejection. In this paper, we propose *double ramp loss* function which gives a continuous upper bound for  $(0 - d - 1)$  loss. Our approach is based on minimizing regularized risk under the double ramp loss using *difference of convex (DC) programming*. We show the effectiveness of our approach through experiments on synthetic and benchmark datasets. Our approach performs better than the state of the art reject option classification approaches.

## 1 Introduction

The primary focus of classification problems has been on algorithms that return a prediction on every example. However, in many real life situations, it may be prudent to *reject* an example rather than run the risk of a costly potential misclassification. Consider, for instance, a physician who has to return a diagnosis for a patient based on the observed symptoms and a preliminary examination. If the symptoms are either ambiguous, or rare enough to be unexplainable without further investigation, then the physician might choose not to risk misdiagnosing the patient (which might lead to further complications). He might instead ask for further medical tests to be performed, or refer the case to an appropriate specialist. Similarly, a banker, when faced with a loan application from a customer, may choose not to decide on the basis of the available information, and ask for a credit bureau score. While the follow-up actions might vary (asking for more features to describe the example, or using a different classifier), the principal response in these cases is to “reject” the example. This paper focuses on the manner in which this principal response is decided, i.e., which examples should a classifier reject, and why? From a geometric standpoint, we can view the classifier as being possessed of a decision surface (which separates points of different classes) as well as a rejection surface. The size of the rejection region impacts the proportion of cases that are likely to be rejected by the classifier, as well as the proportion of predicted cases that are likely to be correctly classified.

A well-optimized classifier with a reject option is the one which minimizes the rejection rate as well as the mis-classification rate on the predicted examples.

Let  $\mathbf{x} \in \mathbb{R}^p$  is the feature vector and  $y \in \{-1, +1\}$  is the class label. Let  $\mathcal{D}(\mathbf{x}, y)$  be the joint distribution of  $\mathbf{x}$  and  $y$ . A typical *reject option classifier* is defined using a bandwidth parameter ( $\rho$ ) and a separating surface ( $f(\mathbf{x}) = 0$ ).  $\rho$  is the parameter which determines the rejection region. Then a reject option classifier  $h(f(\mathbf{x}), \rho)$  is formed as:

$$h(f(\mathbf{x}), \rho) = \begin{cases} 1 & \text{if } f(\mathbf{x}) > \rho \\ 0 & \text{if } |f(\mathbf{x})| \leq \rho \\ -1 & \text{if } f(\mathbf{x}) < -\rho \end{cases} \quad (1)$$

The reject option classifier can be viewed as two parallel surfaces with the rejection area in between. The goal is to determine  $f(\mathbf{x})$  as well as  $\rho$  simultaneously. The performance of this classifier is evaluated using  $L_{0-d-1}$  [13,9] which is

$$L_{0-d-1}(f(\mathbf{x}), y, \rho) = \begin{cases} 1, & \text{if } yf(\mathbf{x}) < -\rho \\ d, & \text{if } |f(\mathbf{x})| \leq \rho \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

In the above loss,  $d$  is the cost of rejection. If  $d = 0$ , then we will always reject. When  $d > .5$ , then we will never reject (because expected loss of random labeling is 0.5). Thus, we always take  $d \in (0, .5)$ .

To learn a reject option classifier, the expectation of  $L_{0-d-1}(\cdot, \cdot, \cdot)$  with respect to  $\mathcal{D}(\mathbf{x}, y)$  (*risk*) is minimized. Since  $\mathcal{D}(\mathbf{x}, y)$  is fixed but unknown, the empirical risk minimization principle is used. The risk under  $L_{0-d-1}$  is minimized by *generalized Bayes discriminant* [9,4], which is as below:

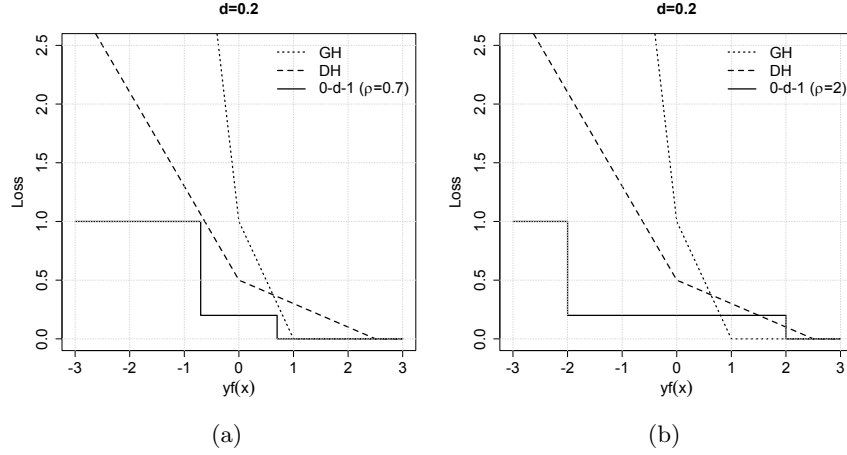
$$f_d^*(\mathbf{x}) = \begin{cases} -1, & \text{if } P(y = 1|\mathbf{x}) < d \\ 0, & \text{if } d \leq P(y = 1|\mathbf{x}) \leq 1 - d \\ 1, & \text{if } P(y = 1|\mathbf{x}) > 1 - d \end{cases} \quad (3)$$

$h(f(\mathbf{x}), \rho)$  (equation (1)) is shown to be infinite sample consistent with respect to the generalized Bayes classifier  $f_d^*(\mathbf{x})$  described in equation (3) [15].

Loss Function	Definition
Generalized Hinge	$L_{GH}(f(\mathbf{x}), y) = \begin{cases} 1 - \frac{1-d}{d}yf(\mathbf{x}), & \text{if } yf(\mathbf{x}) < 0 \\ 1 - yf(\mathbf{x}), & \text{if } 0 \leq yf(\mathbf{x}) < 1 \\ 0, & \text{otherwise} \end{cases}$
Double Hinge	$L_{DH}(f(\mathbf{x}), y) = \max[-y(1-d)f(\mathbf{x}) + H(d), -ydf(\mathbf{x}) + H(d), 0]$ where $H(d) = -d \log(d) - (1-d) \log(1-d)$

**Table 1.** Convex surrogates for  $L_{0-d-1}$ .

Since minimizing the risk under  $L_{0-d-1}$  is computationally cumbersome, convex surrogates for  $L_{0-d-1}$  have been proposed. *Generalized hinge loss*  $L_{GH}$



**Fig. 1.**  $L_{GH}$  and  $L_{DH}$  for  $d = 0.2$ . (a) For  $\rho = 0.7$ , both the losses upper bound the  $L_{0-d-1}$ . For  $\rho = 2$ , both the losses fail to upper bound  $L_{0-d-1}$ .  $L_{GH}$  and  $L_{DH}$  both increase linearly even in the rejection region than being flat.

(see Table 1) is a convex surrogate for  $L_{0-d-1}$  [13,14,3]. It is shown that a minimizer of risk under  $L_{GH}$  is consistent to the generalized Bayes classifier [3]. *Double hinge loss*  $L_{DH}$  (see Table 1) is another convex surrogate for  $L_{0-d-1}$  [7]. Minimizer of the risk under  $L_{DH}$  is shown to be *strongly universally consistent* to the generalized Bayes classifier [7].

We observe that these convex loss functions have some limitations. For example,  $L_{GH}$  is a convex upper bound to  $L_{0-d-1}$  provided  $\rho < 1 - d$  and  $L_{DH}$  forms an upper bound to  $L_{0-d-1}$  provided  $\rho \in (\frac{1-H(d)}{1-d}, \frac{H(d)-d}{d})$  (see Fig. 1). Also, both  $L_{GH}$  and  $L_{DH}$  increase linearly in the rejection region instead of remaining constant. These convex losses can become unbounded for misclassified examples with the scaling of parameters of  $f$ . Moreover, limited experimental results are shown to validate the practical significance of these losses [13,14,3,7]. A non-convex formulation for learning reject option classifier is proposed in [5]. However, theoretical guarantees for the approach proposed in [5] are not known. While learning a reject option classifier, one has to deal with the overlapping class regions as well as the presence of outliers. SVM and other convex loss based approaches are less robust to label noise and outliers in the data [11]. It is shown that ramp loss based risk minimization is more robust to noise [6].

Motivated from this, we propose *double ramp loss* ( $L_{DR}$ ) which incorporates a different loss value for rejection.  $L_{DR}$  forms a continuous nonconvex upper bound for  $L_{0-d-1}$  and overcomes many of the issues of convex surrogates of  $L_{0-d-1}$ . To learn a reject option classifier, we minimize the regularized risk under  $L_{DR}$  which becomes an instance of difference of convex (DC) functions. To minimize such a DC function, we use difference of convex programming approach [1], which essentially solves a sequence of convex programs. The proposed method has following advantages over the existing approaches: (1) the proposed loss function  $L_{DR}$  gives a tighter upper bound to the  $L_{0-d-1}$ , (2)  $L_{DR}$  requires no

constraint on  $\rho$  unlike  $L_{GH}$  and  $L_{DH}$ , (3) our approach can be easily kernelized for dealing with nonlinear problems.

The rest of the paper is organized as follows. In Section 2 we define the *double ramp loss* function ( $L_{DR}$ ) and discuss its properties. Then we discussed the proposed formulation based on risk minimization under  $L_{DR}$ . In Section 3 we derive the algorithm for learning reject option classifier based on regularized risk minimization under ( $L_{DR}$ ) using DC programming. We present experimental results in Section 4. We conclude the paper with the discussion in Section 5.

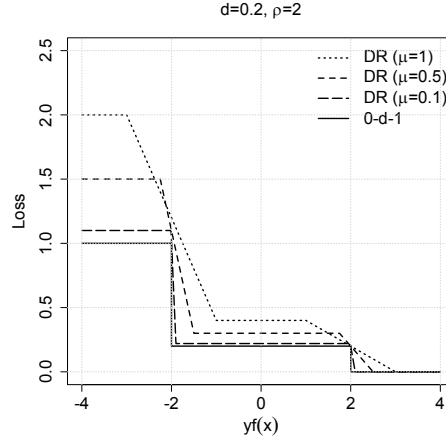
## 2 Proposed Approach

Our approach for learning classifier with reject option is based on minimizing regularized risk under  $L_{DR}$  (double ramp loss).

### 2.1 Double Ramp Loss

We define double ramp loss function as a continuous upper bound for  $L_{0-d-1}$ . This loss function is defined as a sum of two ramp loss functions as follows:

$$L_{DR}(f(\mathbf{x}), y, \rho) = \frac{d}{\mu} \left[ [\mu - yf(\mathbf{x}) + \rho]_+ - [-\mu^2 - yf(\mathbf{x}) + \rho]_+ \right] + \frac{(1-d)}{\mu} \left[ [\mu - yf(\mathbf{x}) - \rho]_+ - [-\mu^2 - yf(\mathbf{x}) - \rho]_+ \right] \quad (4)$$



**Fig. 2.**  $L_{DR}$  and  $L_{0-d-1} : \forall \mu \geq 0, \rho \geq 0$ ,  $L_{DR}$  is an upper bound for  $L_{0-d-1}$ .

where  $[a]_+ = \max(0, a)$ .  $\mu \in (0, 1]$  defines the slope of ramps in the loss function.  $d \in (0, .5)$  is the cost of rejection and  $\rho \geq 0$  is the parameter which defines the size of the rejection region around the classification boundary  $f(\mathbf{x}) =$

0.<sup>4</sup> As in  $L_{0-d-1}$ ,  $L_{DR}$  also considers the region  $[-\rho, \rho]$  as rejection region. Fig. 2 shows  $L_{DR}$  for  $d = 0.2, \rho = 2$  with different values of  $\mu$ .

**Theorem 1.** (1)  $L_{DR} \geq L_{0-d-1}$ ,  $\forall \mu > 0, \rho \geq 0$ . (2)  $\lim_{\mu \rightarrow 0} L_{DR}(f(\mathbf{x}), \rho, y) = L_{0-d-1}(f(\mathbf{x}), \rho, y)$ . (3) In the rejection region  $yf(\mathbf{x}) \in (\rho - \mu^2, -\rho + \mu)$ , the loss remains constant, that is  $L_{DR}(f(\mathbf{x}), y, \rho) = d(1 + \mu)$ . (4) For  $\mu > 0$ ,  $L_{DR} \leq (1 + \mu)$ ,  $\forall \rho \geq 0, \forall d \geq 0$ . (5) When  $\rho = 0$ ,  $L_{DR}$  is same as  $\mu$ -ramp loss ([12]) used for classification problems without rejection option. (6)  $L_{DR}$  is a non-convex function of  $(yf(\mathbf{x}), \rho)$ .

The proof of Theorem 1 is provided in Appendix A. We see that  $L_{DR}$  does not put any restriction on  $\rho$  for it to be an upper bound of  $L_{0-d-1}$ . Thus,  $L_{DR}$  is a general ramp loss function which also allows rejection option.

## 2.2 Risk Formulation Using $L_{DR}$

Let  $\mathcal{S} = \{(\mathbf{x}_n, y_n), n = 1 \dots N\}$  be the training dataset, where  $\mathbf{x}_n \in \mathbb{R}^p$ ,  $y_n \in \{-1, +1\}$ ,  $\forall n$ . As discussed, we minimize regularized risk under  $L_{DR}$  to find a reject option classifier. In this paper, we use  $l_2$  regularization. Let  $\Theta = [\mathbf{w}^T \ b \ \rho]^T$ . Thus, for  $f(\mathbf{x}) = (\mathbf{w}^T \phi(\mathbf{x}) + b)$ , regularized risk under double ramp loss is

$$\begin{aligned} R(\Theta) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N L_{DR}(y_n, \mathbf{w}^T \phi(\mathbf{x}_n) + b) \\ &= \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N \left\{ d[\mu - y_n f(\mathbf{x}_n) + \rho]_+ - d[-\mu^2 - y_n f(\mathbf{x}_n) + \rho]_+ \right. \\ &\quad \left. + (1-d)[\mu - y_n f(\mathbf{x}_n) - \rho]_+ - (1-d)[- \mu^2 - y_n f(\mathbf{x}_n) - \rho]_+ \right\} \\ &= \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N \left\{ d[\mu - y_n f(\mathbf{x}_n) + \rho]_+ + (1-d)[\mu - y_n f(\mathbf{x}_n) - \rho]_+ \right. \\ &\quad \left. - d[-\mu^2 - y_n f(\mathbf{x}_n) + \rho]_+ - (1-d)[- \mu^2 - y_n f(\mathbf{x}_n) - \rho]_+ \right\} \end{aligned}$$

where  $C$  is regularization parameter. While minimizing  $R(\Theta)$ , no non-negativity condition on  $\rho$  is required due to the following lemma.

**Lemma 1.** At the minimum of  $R(\Theta)$ ,  $\rho$  must be non-negative.

Proof of the above lemma is provided in Appendix B.

<sup>4</sup> While  $L_{DR}$  is parametrized by  $\mu$  and  $d$  as well, we omit them for the sake of notational consistency.

### 3 Solution methodology

$R(\Theta)$  (equation (5)) is a nonconvex function of  $\Theta$ . However,  $R(\Theta)$  can be written as  $R(\Theta) = R_1(\Theta) - R_2(\Theta)$ , where  $R_1(\Theta)$  and  $R_2(\Theta)$  are convex functions of  $\Theta$ .

$$R_1(\Theta) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N \left[ d[\mu - y_n f(\mathbf{x}_n) + \rho]_+ + (1-d)[\mu - y_n f(\mathbf{x}_n) - \rho]_+ \right]$$

$$R_2(\Theta) = \frac{C}{\mu} \sum_{n=1}^N \left[ d[-\mu^2 - y_n f(\mathbf{x}_n) + \rho]_+ + (1-d)[- \mu^2 - y_n f(\mathbf{x}_n) - \rho]_+ \right]$$

In this case, DC programming guarantees to find a local optima of  $R(\Theta)$  [1]. In the simplified DC algorithm [1], an upper bound of  $R(\Theta)$  is found using the convexity property of  $R_2(\Theta)$  as follows.

$$R(\Theta) \leq R_1(\Theta) - R_2(\Theta^{(l)}) - (\Theta - \Theta^{(l)})^T \nabla R_2(\Theta^{(l)}) =: ub(\Theta, \Theta^{(l)}) \quad (5)$$

where  $\Theta^{(l)}$  is the parameter vector after  $(l)^{th}$  iteration,  $\nabla R_2(\Theta^{(l)})$  is a sub-gradient of  $R_2$  at  $\Theta^{(l)}$ .  $\Theta^{(l+1)}$  is found by minimizing  $ub(\Theta, \Theta^{(l)})$ . Thus,  $R(\Theta^{(l+1)}) \leq ub(\Theta^{(l+1)}, \Theta^{(l)}) \leq ub(\Theta^{(l)}, \Theta^{(l)}) = R(\Theta^{(l)})$ . Which means, in every iteration, the DC program reduces the value of  $R(\Theta)$ .

#### 3.1 Learning Reject Option Classifier Using DC Programming

In this section, we will derive a DC algorithm for minimizing  $R(\Theta)$ . We initialize with  $\Theta = \Theta^{(0)}$ . For any  $l \geq 0$ , we find  $ub(\Theta, \Theta^{(l)})$  as an upper bound for  $R(\Theta)$  (see equation (5)) as follows:

$$ub(\Theta, \Theta^{(l)}) = R_1(\Theta) - R_2(\Theta^{(l)}) - (\Theta - \Theta^{(l)})^T \nabla R_2(\Theta^{(l)})$$

Given  $\Theta^{(l)}$ , we find  $\Theta^{(l+1)}$  by minimizing the upper bound  $ub(\Theta, \Theta^{(l)})$ . Thus,

$$\Theta^{(l+1)} \in \arg \min_{\Theta} ub(\Theta, \Theta^{(l)}) = \arg \min_{\Theta} R_1(\Theta) - \Theta^T \nabla R_2(\Theta^{(l)}) \quad (6)$$

where  $\nabla R_2(\Theta^{(l)})$  is the subgradient of  $R_2(\Theta)$  at  $\Theta^{(l)}$ . We choose  $\nabla R_2(\Theta^{(l)})$  as:

$$\nabla R_2(\Theta^{(l)}) = \sum_{n=1}^N \beta_n'^{(l)} [-y_n \phi(\mathbf{x}_n)^T \quad -y_n \quad 1]^T + \sum_{n=1}^N \beta_n''^{(l)} [-y_n \phi(\mathbf{x}_n)^T \quad -y_n \quad -1]^T$$

where

$$\begin{cases} \beta_n'^{(l)} = \frac{Cd}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) - \rho^{(l)} < -\mu^2\}} \\ \beta_n''^{(l)} = \frac{C(1-d)}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) + \rho^{(l)} < -\mu^2\}} \end{cases} \quad (7)$$

For  $f(\mathbf{x}) = (\mathbf{w}^T \phi(\mathbf{x}) + b)$ , we rewrite the upper bound minimization problem described in equation (6) as follows,

$$\begin{aligned} P^{(l+1)} &= \min_{\Theta} R_1(\Theta) - \Theta^T \nabla R_2(\Theta^{(l)}) \\ &= \min_{\mathbf{w}, b, \rho} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N \left[ d[\mu - y_n f(\mathbf{x}_n) + \rho]_+ + (1-d)[\mu - y_n f(\mathbf{x}_n) - \rho]_+ \right] \\ &\quad + \sum_{n=1}^N \beta_n'^{(l)} [y_n f(\mathbf{x}_n) - \rho] + \sum_{n=1}^N \beta_n''^{(l)} [y_n f(\mathbf{x}_n) + \rho] \end{aligned}$$

Note that  $P^{(l+1)}$  is a convex optimization problem where the optimization variables are  $(\mathbf{w}, b, \rho)$ . We rewrite  $P^{(l+1)}$  as

$$\begin{aligned} P^{(l+1)} &= \min_{\mathbf{w}, b, \xi', \xi''} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N [d\xi_n' + (1-d)\xi_n''] + \sum_{n=1}^N \beta_n'^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \rho] \\ &\quad + \sum_{n=1}^N \beta_n''^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] \\ s.t. \quad &y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq \rho + \mu - \xi_n', \quad \xi_n' \geq 0, \quad n = 1 \dots N \\ &y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq -\rho + \mu - \xi_n'', \quad \xi_n'' \geq 0 \quad n = 1 \dots N \end{aligned}$$

where  $\xi' = [\xi_1' \ \xi_2' \dots \xi_N']^T$  and  $\xi'' = [\xi_1'' \ \xi_2'' \dots \xi_N'']^T$ . The dual optimization problem  $D^{(l+1)}$  of  $P^{(l+1)}$  is as follows.

$$\begin{aligned} D^{(l+1)} &= \min_{\gamma', \gamma''} \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N y_n y_m (\gamma_n' + \gamma_n'') (\gamma_m' + \gamma_m'') k(\mathbf{x}_n, \mathbf{x}_m) - \mu \sum_{n=1}^N (\gamma_n' + \gamma_n'') \\ s.t. \quad &\begin{cases} -\beta_n'^{(l)} \leq \gamma_n' \leq \frac{Cd}{\mu} - \beta_n'^{(l)} & n = 1 \dots N \\ -\beta_n''^{(l)} \leq \gamma_n'' \leq \frac{C(1-d)}{\mu} - \beta_n''^{(l)} & n = 1 \dots N \\ \sum_{n=1}^N y_n (\gamma_n' + \gamma_n'') = 0 & \sum_{n=1}^N (\gamma_n' - \gamma_n'') = 0 \end{cases} \end{aligned}$$

where  $\gamma' = [\gamma_1' \ \gamma_2' \dots \gamma_N']^T$  and  $\gamma'' = [\gamma_1'' \ \gamma_2'' \dots \gamma_N'']^T$  are dual variables. The derivation of dual  $D^{(l+1)}$  can be seen in Appendix C. At the optimality of  $P^{(l+1)}$ ,  $\mathbf{w}$  can be found as  $\mathbf{w} = \sum_{n=1}^N y_n (\gamma_n' + \gamma_n'') \phi(\mathbf{x}_n)$ .

Since  $P^{(l+1)}$  has quadratic objective and linear constraints, it holds strong duality with  $D^{(l+1)}$ . Solving  $D^{(l+1)}$  is more useful as it can be easily kernelized for non-linear problems. Behavior of  $\gamma_n'$  and  $\gamma_n''$  under different cases is as follows.

$$\begin{cases} y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \mu > \rho & \Rightarrow \gamma_n' = -\beta_n'^{(l)}; \quad \gamma_n'' = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \mu = \rho & \Rightarrow \gamma_n' \in (-\beta_n'^{(l)}, \frac{Cd}{\mu} - \beta_n'^{(l)}); \quad \gamma_n'' = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \mu \in (-\rho, \rho) & \Rightarrow \gamma_n' = \frac{Cd}{\mu} - \beta_n'^{(l)}; \quad \gamma_n'' = -\beta_n''^{(l)} \\ y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \mu = -\rho & \Rightarrow \gamma_n' = \frac{Cd}{\mu} - \beta_n'^{(l)}; \quad \gamma_n'' \in (-\beta_n''^{(l)}, \frac{C(1-d)}{\mu} - \beta_n''^{(l)}) \\ y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \mu < -\rho & \Rightarrow \gamma_n' = \frac{Cd}{\mu} - \beta_n'^{(l)}; \quad \gamma_n'' = \frac{C(1-d)}{\mu} - \beta_n''^{(l)} \end{cases}$$

### 3.2 Finding $b^{(l+1)}$ and $\rho^{(l+1)}$

The dual optimization problem above gives dual variables  $\gamma'^{(l+1)}$  and  $\gamma''^{(l+1)}$  using which the normal vector is found as  $\mathbf{w}^{(l+1)} = \sum_{n=1}^N (\gamma_n'^{(l+1)} + \gamma_n''^{(l+1)}) y_n \phi(\mathbf{x}_n)$ . To find  $b^{(l+1)}$  and  $\rho^{(l+1)}$ , we consider  $\mathbf{x}_n \in \text{SV}'^{(l+1)} \cup \text{SV}''^{(l+1)}$ , where

$$\begin{aligned} \text{SV}'^{(l+1)} &= \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l+1)} + b^{(l+1)}) = \rho^{(l+1)} + \mu\} \\ \text{SV}''^{(l+1)} &= \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l+1)} + b^{(l+1)}) = -\rho^{(l+1)} + \mu\} \end{aligned}$$

We already saw that

1. If  $\mathbf{x}_n \in \text{SV}'^{(l+1)}$ , then  $\gamma_n'^{(l+1)} \in (-\beta_n'^{(l)}, \frac{Cd}{\mu} - \beta_n'^{(l)})$  and  $\gamma_n''^{(l+1)} = -\beta_n''^{(l)}$
2. If  $\mathbf{x}_n \in \text{SV}''^{(l+1)}$ , then  $\gamma_n'^{(l+1)} = \frac{Cd}{\mu} - \beta_n'^{(l)}$  and  $\gamma_n''^{(l+1)} \in (-\beta_n''^{(l)}, \frac{C(1-d)}{\mu} - \beta_n''^{(l)})$

We solve the system of linear equations corresponding to sets  $\text{SV}'^{(l+1)}$  and  $\text{SV}''^{(l+1)}$  for identifying  $b^{(l+1)}$  and  $\rho^{(l+1)}$ .

### 3.3 Summary of the Algorithm

We fix  $d \in [0, .5]$ ,  $\mu \in (0, 1]$  and  $C$  and initialize the parameter vector  $\Theta$  as  $\Theta^{(0)}$ . In any iteration ( $l$ ), we find  $\beta_n'^{(l)}, \beta_n''^{(l)}$ ,  $n = 1 \dots N$  (see equation (7)) using  $\Theta^{(l)}$ . We use  $\beta_n'^{(l)}, \beta_n''^{(l)}$ ,  $n = 1 \dots N$  and solve  $D^{(l+1)}$  to find  $\gamma'^{(l+1)}, \gamma''^{(l+1)}$ .  $\mathbf{w}^{(l+1)}$  is found as  $\mathbf{w}^{(l+1)} = \sum_{n=1}^N y_n (\gamma_n'^{(l+1)} + \gamma_n''^{(l+1)}) \phi(\mathbf{x}_n)$ . We find  $b^{(l+1)}$  and  $\rho^{(l+1)}$  as described in Section 3.2. Thus, we have found  $\Theta^{(l+1)}$ . Using  $\Theta^{(l+1)}$ , we now find  $\beta_n'^{(l+1)}, \beta_n''^{(l+1)}$ ,  $n = 1 \dots N$ . We repeat the above two steps until the parameter vector  $\Theta$  changes significantly. More formal description of our algorithm is provided in Algorithm 1.

### 3.4 $\gamma'$ and $\gamma''$ at the Convergence of Algorithm 1

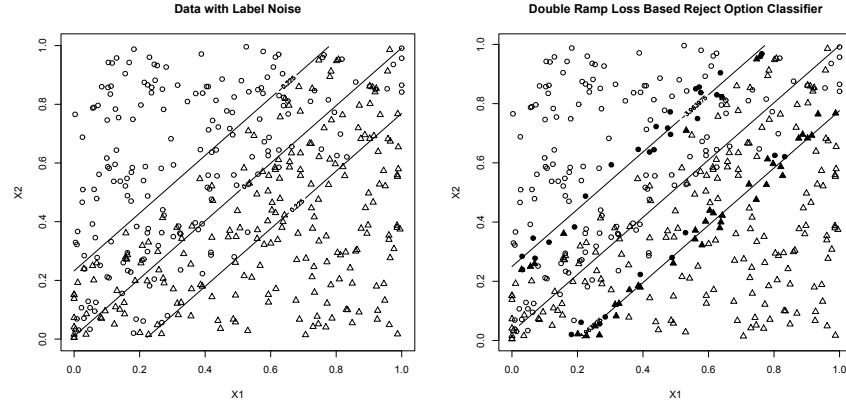
At the convergence of Algorithm 1, let  $\gamma_n'^*, \gamma_n''^*$ ,  $n = 1 \dots N$  become the values of the dual variables. The behavior of  $\gamma_n'^*$  and  $\gamma_n''^*$  is described in Table 2. For any  $\mathbf{x}_n$ , only one of  $\gamma_n'^*$  and  $\gamma_n''^*$  can be nonzero. We observe that parameters  $\mathbf{w}, b$  and  $\rho$  are determined by the points whose margin ( $y f(\mathbf{x})$ ) is in the range  $[\rho - \mu^2, \rho + \mu] \cup [-\rho - \mu^2, -\rho + \mu]$ . We call these points as *support vectors*. We also see that  $\mathbf{x}_n$  for which  $y_n f(\mathbf{x}_n) \in (\rho + \mu, \infty) \cup (-\rho + \mu, \rho - \mu^2) \cup (-\infty, -\rho - \mu^2)$ , both  $\gamma_n'^*, \gamma_n''^* = 0$ . Thus, points which are correctly classified with margin at least  $(\rho + \mu)$ , points falling close to the decision boundary with margin in the interval  $(-\rho + \mu, \rho - \mu^2)$  and points misclassified with a high negative margin (less than  $-\rho - \mu^2$ ), are ignored in the final classifier. Thus, our approach not only rejects points falling in the overlapping region of classes, it also ignores potential outliers. We illustrate these insights through experiments on a synthetic dataset as shown in Fig. 3. 400 points are uniformly sampled from the square region  $[0 \ 1] \times [0 \ 1]$ . We consider the diagonal passing through the origin as the separating surface



**Algorithm 1** Learning Reject Option Classifier by Minimizing  $R(\Theta)$ **Input :**  $d \in [0, .5]$ ,  $\mu \in (0, 1]$ ,  $C > 0$ ,  $\mathcal{S}$ **Output :**  $\mathbf{w}^*, b^*, \rho^*$ **Initialize**  $\mathbf{w}^{(0)}, b^{(0)}, \rho^{(0)}$ ,  $l = 0$ **repeat**    **Compute**  $\beta_n^{(l)} = \frac{Cd}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) - \rho^{(l)} < -\mu^2\}}$      $\beta_n^{\prime(l)} = \frac{C(1-d)}{\mu} \mathbb{I}_{\{y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l)} + b^{(l)}) + \rho^{(l)} < -\mu^2\}}$     **Find**  $\gamma^{\prime(l+1)}, \gamma^{\prime\prime(l+1)}$  by solving  $D^{(l+1)}$  described in equation (8)    **Find**  $\mathbf{w}^{(l+1)} = \sum_{n=1}^N y_n(\gamma_n^{\prime(l+1)} + \gamma_n^{\prime\prime(l+1)})\phi(\mathbf{x}_n)$     **Find**  $b^{(l+1)}$  and  $\rho^{(l+1)}$  by solving the system of linear equations corresponding to sets  $SV_1^{(l+1)}$  and  $SV_2^{(l+1)}$ , where

$$SV_1^{\prime(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l+1)} + b^{(l+1)}) = \rho^{(l+1)} + \mu\}$$

$$SV_2^{\prime(l+1)} = \{\mathbf{x}_n \mid y_n(\phi(\mathbf{x}_n)^T \mathbf{w}^{(l+1)} + b^{(l+1)}) = -\rho^{(l+1)} + \mu\}$$

**until** convergence of  $\Theta^{(l)}$ 

**Fig. 3.** Figure on left shows that label noise affects points near the true classification boundary. Classes are represented using empty *circles* and *triangles*. Figure on right shows reject option classifier learnt using the proposed  $L_{DR}$  based approach ( $C = 100$ ,  $\mu = 1$ ,  $d = .2$ ). Filled *circles* and *triangles* represent the support vectors.

Condition	$\gamma_n^{I*} \in$	$\gamma_n^{II*} \in$
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \in (\rho + \mu, \infty)$	0	0
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) = \rho + \mu$	$(0, \frac{Cd}{\mu})$	0
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \in [\rho - \mu^2, \rho + \mu)$	$\frac{Cd}{\mu}$	0
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \in (-\rho + \mu, \rho - \mu^2)$	0	0
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) = -\rho + \mu$	0	$(0, \frac{C(1-d)}{\mu})$
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \in [-\rho - \mu^2, -\rho + \mu)$	0	$\frac{C(1-d)}{\mu}$
$y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \in (-\infty, -\rho - \mu^2)$	0	0

**Table 2.** Behavior of  $\gamma^{I*}$  and  $\gamma^{II*}$ 

and assign labels  $\{-1, +1\}$  to all the points using it. We changed the labels of 80 points inside the band (width=0.225) around the separating surface. Fig. 3 shows the reject option classifier learnt using the proposed method. We see that the proposed approach learns the rejection region accurately. We also observe that all of the support vectors are near the two parallel hyperplanes.

## 4 Experimental Results

We show the effectiveness of our approach by showing its performance on several datasets. We also compare our approach with the approach proposed in [7].

### 4.1 Dataset Description

We report experimental results on 1 synthetic datasets and 2 datasets taken from UCI ML repository [2].

1. **Synthetic Dataset 1 :** Let  $f_1$  and  $f_2$  be two mixture density functions in  $\mathbb{R}^2$  defined as follows:

$$f_1(\mathbf{x}) = 0.45\mathcal{U}([1, 0] \times [1, 1]) + 0.5\mathcal{U}([4, 3] \times [0, 1]) + 0.05\mathcal{U}([10, 0] \times [5, 5])$$

$$f_2(\mathbf{x}) = 0.45\mathcal{U}([0, 1] \times [1, 1]) + 0.5\mathcal{U}([9, 10] \times [1, 0]) + 0.05\mathcal{U}([0, 10] \times [5, 5])$$

where  $\mathcal{U}(A)$  denotes the uniform density function with support set  $A$ . We sample 150 points independently each from  $f_1$  and  $f_2$ . We label these points using the hyperplane with  $\mathbf{w} = [1 \ 0]^T$  and  $b = 0$ . We choose 10% of these points uniformly at random and flip their labels.

2. **Synthetic Dataset 2 [8] :**  $\mathbf{m}_{k1}, k = 1, \dots, 10$  were drawn from  $\mathcal{N}((1, 0)^T, I)$  and labeled as class  $C_1$ . Similarly,  $\mathbf{m}_{k2}, k = 1, \dots, 10$  were drawn from  $\mathcal{N}((0, 1)^T, I)$  and labeled as class  $C_2$ . For each class, 100 observations were drawn from the following mixture distributions:

$$f(\mathbf{x}|C_i) = \sum_{k=1}^{10} \frac{1}{10} \mathcal{N}(\mathbf{m}_{ki}, I/5), \quad i = 1, 2$$

3. **Ionosphere Dataset [2] :** This dataset describes the problem of discriminating *good versus bad radars* based on whether they send some useful information about the Ionosphere. There are 34 variables and 351 observations.

4. **Parkinsons Disease Dataset [2]** : This dataset is used to discriminate people with Parkinsons disease from the healthy people. There are 22 features which are comprised of a range of biomedical voice measurements from individuals. There are 195 such feature vectors.

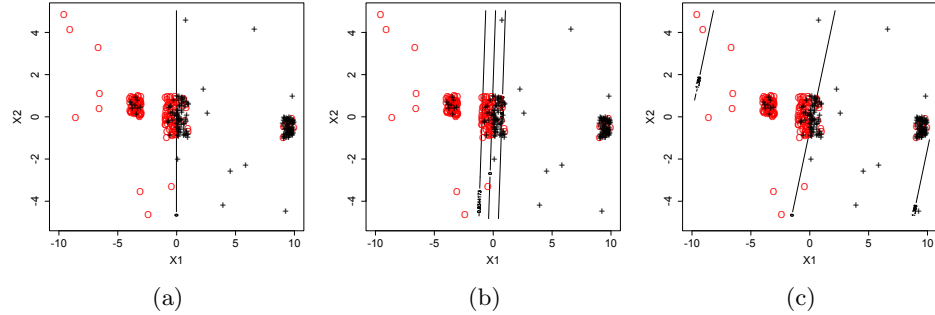
## 4.2 Experimental Setup

In the proposed  $L_{DR}$  based approach, for solving the dual  $D^{(l)}$  at every iteration, we have used the *kernelab* package [10] in **R**. We thank the authors of  $L_{DH}$  based method [7] for providing the codes for their approach. For nonlinear problems, we use RBF kernel. In our approach, we set  $\mu = 1$ .  $C$  and  $\sigma$  (width parameter for RBF kernel) are chosen using 10-fold cross validation.

## 4.3 Simulation Results

For every dataset, we report results for values of  $d$  in the interval  $[0.05 \dots .5]$  with the step size of 0.05. For every value of  $d$ , we find the cross validation risk (under  $L_{0-d-1}$ ), % accuracy on the non-rejected examples (Acc) and % rejection rate (RR). The results provided are based on 10 repetitions of 10-fold cross validation (CV). We show the average values and standard deviation (computed over the 10 repetitions).

We now discuss the experimental results. Fig. 4(a) shows the Synthetic dataset and the true classification boundary. This dataset has some mislabeled points creating noise around the classification surface. Fig. 4(b) and (c) show the classifiers learnt using  $L_{DR}$  and  $L_{DH}$  based approaches respectively for  $d = 0.2$ . We see that  $L_{DR}$  based approach accurately finds the true classification boundary as oppose to  $L_{DH}$  based approach. Also, the reject region found by  $L_{DR}$  based approach is covers the most ambiguous region unlike  $L_{DH}$  based approach which rejects almost all the points.



**Fig. 4.** (a) Synthetic Dataset and the true classification boundary. Reject option classifiers learnt using (b) proposed  $L_{DR}$  based approach for  $d = 0.2$ , (c)  $L_{DH}$  based approach for  $d = 0.2$ .

d	$L_{DR} (C = 2)$			$L_{DH} (C = 32)$		
	Risk	RR	Acc on un-rejected	Risk	RR	Acc on un-rejected
0.05	0.068±0.015	90.87±5.79	75.87±7.95	<b>0.05</b>	100	NA
0.1	0.138±0.023	70.35±12.18	79.05±6.87	<b>0.105</b> ±0.002	95.53±1.69	77.20±6.06
0.15	<b>0.135</b> ±0.003	65.41±5.06	89.66±0.90	0.136	72.77±0.23	90.56±0.66
0.2	<b>0.155</b> ±0.006	43.18±4.31	88.56±0.75	0.17	72.67	90.36±1.44
0.25	<b>0.164</b> ±0.014	32.13±8.43	87.97±1.42	0.204±0.003	66.5±1.7	91±0.74
0.3	<b>0.148</b> ±0.012	13.23±7.52	87.67±0.69	0.197	46.73±0.14	89.37±0.32
0.35	<b>0.134</b> ±0.005	4.57±1.80	87.68±0.23	0.21±0.002	43.33±0.65	90.02±0.38
0.4	<b>0.131</b> ±0.003	1.51±0.56	87.29±0.30	0.21±0.006	31.17±1.26	87.41±0.55
0.45	<b>0.128</b> ±0.002	0.86±0.45	87.45±0.25	0.265±0.008	9.13±1.1	75.58±0.98
0.5	<b>0.136</b> ±0.01	0	86.41±0.99	0.297±0.004	0	70.27±0.44

**Table 3.** Comparison results on Synthetic Dataset 1 (linear classifiers for both the approaches).

Table 3-6 show the experimental results on all the datasets. We observe the following:

1. We see that the proposed  $L_{DR}$  based method outperforms  $L_{DH}$  based approach in terms of the risk (expectation of  $L_{0-d-1}$ ). For Synthetic dataset 1, except for  $d = 0.05$  and  $0.1$ ,  $L_{DR}$  based method has lower CV risk. For Synthetic dataset 2, both the approaches perform comparable to each other. For Ionosphere dataset, except for  $d = 0.2, 0.25$  and  $0.3$ ,  $L_{DR}$  based method has lower CV risk. For Parkinsons dataset,  $L_{DR}$  based method has lower CV risk except for  $d = 0.35$ .
2. We also observe that  $L_{DR}$  based method outputs classifiers with significantly lesser rejection rate for all the datasets and for all values of  $d$ .

Thus, for most of the cases, the proposed  $L_{DR}$  based approach outputs classifiers with lesser risk. Moreover, the learnt classifier has always lesser rejection rate compared to the  $L_{DH}$  based approach.

## 5 Conclusion and Future Work

In this paper, we have proposed a new loss function  $L_{DR}$  (**double ramp loss**) for learning the reject option classifier.  $L_{DR}$  gives tighter upper bound for  $L_{0-d-1}$  compared to convex losses  $L_{DH}$  and  $L_{GH}$ . Our approach learns the classifier by minimizing the regularized *risk* under the double ramp loss function which becomes an instance of DC optimization problem. Our approach can also learn nonlinear classifiers by using appropriate kernel function. Experimentally we have shown that our approach works superior to  $L_{DH}$  based approach for learning reject option classifier.

d	$L_{DR} (C = 64, \gamma = 0.25)$			$L_{DH} (C = 64, \gamma = 0.25)$		
	Risk	RR	Acc on un-rejected	Risk	RR	Acc on un-rejected
0.05	0.046±0.006	79.5±1.47	97.56±2.92	0.046±0.004	86.5±0.82	97.26±3.8
0.1	<b>0.096</b> ±0.006	75.45±1.12	92.80±2.35	0.1±0.005	76.35±1.13	91.65±2.0
0.15	0.15±0.012	64.3±2.32	86.40±2.35	<b>0.139</b> ±0.01	52.3±2.02	87.6±2.4
0.2	0.182±0.01	51.2±1.90	84.79±1.99	<b>0.162</b> ±0.007	40.35±1.68	86.75±1.22
0.25	0.193±0.008	30.3±1.01	83.56±1.33	<b>0.18</b> ±0.008	31.25±1.65	85.74±1.47
0.3	0.190±0.005	16.4±1.74	83.47±0.75	<b>0.183</b> ±0.013	18.35±2.85	84.4±1.2
0.35	0.178±0.006	6.85±1.43	83.49±0.69	0.178±0.008	10.65±1.42	84.21±0.80
0.4	<b>0.171</b> ±0.012	2.6±1.26	83.51±1.2	0.177±0.006	5.75±0.68	83.75±0.76
0.45	<b>0.168</b> ±0.011	0.65±0.41	83.42±1.06	0.182±0.008	2.95±0.9	82.61±0.87
0.5	<b>0.178</b> ±0.014	0	82.2±1.36	0.184±0.009	0	81.65±0.88

**Table 4.** Comparison Results on Synthetic Dataset 2 (nonlinear classifiers using RBF kernel for both the approaches).

d	$L_{DR} (C = 2, \gamma = 0.125)$			$L_{DH} (C = 16, \gamma = 0.125)$		
	Risk	RR	Acc on un-rejected	Risk	RR	Acc on un-rejected
0.05	<b>0.025</b> ±0.002	34.84±0.92	98.94±0.31	0.029	52.61±0.73	99.47±0.06
0.1	<b>0.027</b> ±0.003	8.81±0.32	97.99±0.33	0.047±0.002	43.44±0.85	99.46±0.17
0.15	<b>0.039</b> ±0.003	5.78±0.57	96.81±0.29	0.042±0.003	24.02±1.62	99.3±0.37
0.2	0.044±0.001	3.46±0.51	96.18±0.15	<b>0.04</b> ±0.002	17.43±0.59	99.42±0.25
0.25	0.047±0.002	1.76±0.41	95.68±0.23	<b>0.046</b> ±0.001	14.47±0.79	98.9±0.16
0.3	0.052±0.003	0.92±0.46	95.08±0.35	<b>0.051</b> ±0.003	12.57±0.75	98.56±0.31
0.35	<b>0.051</b> ±0.003	0.03±0.09	94.88±0.29	0.054±0.002	9.33±0.59	97.72±0.21
0.4	<b>0.051</b> ±0.002	0	94.95±0.24	0.054±0.003	6.72±0.86	97.09±0.35
0.45	<b>0.054</b> ±0.002	0	94.64±0.21	0.055±0.003	3.53±0.41	95.97±0.36
0.5	<b>0.054</b> ±0.001	0	94.62±0.13	0.055±0.005	0	94.55±0.47

**Table 5.** Comparison results on Ionosphere dataset (nonlinear classifiers using RBF kernel for both the approaches).

## References

1. Le Thi Hoai An and Pham Dinh Tao. Solving a class of linearly constrained indefinite quadratic problems by d.c. algorithms. *Journal of Global Optimization*, 11:253–285, 1997.
2. K. Bache and M. Lichman. UCI machine learning repository, 2013.
3. Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, June 2008.
4. C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, January 1970.
5. Giorgio Fumera and Fabio Roli. Support vector machines with embedded reject option. In *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, SVM '02, pages 68–82, 2002.
6. Aritra Ghosh, Naresh Manwani, and P. S. Sastry. Making risk minimization tolerant to label noise. *CoRR*, abs/1403.3610, 2014.

d	$L_{DR} (C = 32)$			$L_{DH} (C = 32)$		
	Risk	RR	Acc on un-rejected	Risk	RR	Acc on un-rejected
0.05	<b>0.031</b> $\pm 0.002$	43.88 $\pm 0.80$	98.33 $\pm 0.49$	0.043 $\pm 0.001$	86.38 $\pm 0.92$	100
0.1	<b>0.051</b> $\pm 0.004$	41.79 $\pm 0.77$	98.07 $\pm 1.03$	0.061 $\pm 0.002$	53.76 $\pm 1.64$	98.61 $\pm 0.62$
0.15	<b>0.071</b> $\pm 0.002$	40.08 $\pm 1.21$	98.14 $\pm 0.48$	0.086 $\pm 0.004$	39.56 $\pm 1.13$	95.8 $\pm 0.72$
0.2	<b>0.095</b> $\pm 0.004$	37.67 $\pm 1.04$	96.99 $\pm 0.55$	0.125 $\pm 0.008$	29.78 $\pm 2.06$	90.86 $\pm 1.5$
0.25	<b>0.133</b> $\pm 0.009$	20.46 $\pm 2.79$	90.26 $\pm 1.30$	0.142 $\pm 0.004$	22.3 $\pm 1.95$	89.02 $\pm 0.73$
0.3	<b>0.129</b> $\pm 0.01$	4.06 $\pm 2.06$	87.83 $\pm 1.15$	0.131 $\pm 0.009$	14.19 $\pm 1.05$	89.76 $\pm 1.01$
0.35	0.134 $\pm 0.007$	2.49 $\pm 1.04$	87.19 $\pm 0.76$	<b>0.133</b> $\pm 0.004$	9.97 $\pm 1.18$	89.10 $\pm 0.57$
0.4	<b>0.131</b> $\pm 0.008$	0.56 $\pm 0.44$	87.06 $\pm 0.75$	0.133 $\pm 0.006$	6.10 $\pm 1.62$	88.53 $\pm 0.92$
0.45	<b>0.133</b> $\pm 0.013$	0.05 $\pm 0.17$	86.72 $\pm 1.28$	0.14 $\pm 0.009$	2.92 $\pm 1.09$	86.96 $\pm 1.05$
0.5	<b>0.133</b> $\pm 0.009$	0	86.65 $\pm 0.94$	0.139 $\pm 0.008$	0	86.06 $\pm 0.76$

**Table 6.** Comparison results on Parkinsons Disease dataset (linear classifiers for both the approaches).

7. Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support vector machines with a reject option. In *NIPS*, pages 537–544, 2008.
8. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Series of Statistics. New York, N. Y. Springer, 2nd edition, 2009.
9. Radu Herbei and Marten H. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics*, 34(4):709–721, December 2006.
10. Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, November 2004.
11. Naresh Manwani and P. S. Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Systems, Man and Cybernetics: Part-B*, 43:1146–1151, March 2013.
12. Cheng Soon Ong and Le Thi Hoai An. Learning sparse classifiers with difference of convex functions algorithms. *Optimization Methods and Software*, (ahead-of-print):1–25, 2012.
13. Marten Wegkamp and Ming Yuan. Support vector machines with a reject option. *Bernaulli*, 17(4):1368–1385, 2011.
14. Marten H. Wegkamp. Lasso type classifiers with a reject option. *Electronic Journal of Statistics*, 1:155–168, 2007.
15. Ming Yuan and Marten Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11:111–130, March 2010.

## A Proof of Theorem 1

$$\begin{aligned}
L_{DR}(f(\mathbf{x}), \rho, y) &= \frac{d}{\mu} \left[ [\mu - yf(\mathbf{x}) + \rho]_+ - [-\mu^2 - yf(\mathbf{x}) + \rho]_+ \right] \\
&\quad + \frac{(1-d)}{\mu} \left[ [\mu - yf(\mathbf{x}) - \rho]_+ - [-\mu^2 - yf(\mathbf{x}) - \rho]_+ \right]
\end{aligned}$$

Interval	$L_{DR}$	$L_{0-d-1}$
$yf(\mathbf{x}) \in [\rho + \mu, \infty)$	0	0
$yf(\mathbf{x}) \in (\rho, \rho + \mu)$	$\in (0, d)$	0
$yf(\mathbf{x}) \in (\rho - \mu^2, \rho]$	$\in [d, (1 + \mu)d)$	$d$
$yf(\mathbf{x}) \in [-\rho + \mu, \rho - \mu^2]$	$(1 + \mu)d$	$d$
$yf(\mathbf{x}) \in [-\rho, -\rho + \mu)$	$\in ((1 + \mu)d, (1 + \mu)d + (1 - d)]$	$d$
$yf(\mathbf{x}) \in (-\rho - \mu^2, -\rho)$	$\in ((1 + \mu)d + (1 - d), (1 + \mu))$	1
$yf(\mathbf{x}) \in (-\infty, -\rho - \mu^2]$	$1 + \mu$	1

**Table 7.** Proof for Theorem 1.(1).

1. Table 7 shows that  $L_{DR} \geq L_{0-d-1}$ ,  $\forall \mu > 0, \rho \geq 0$ .
2. We need to show that  $\lim_{\mu \rightarrow 0} L_{DR}(f(\mathbf{x}), \rho, y) = L_{0-d-1}(f(\mathbf{x}), \rho, y)$ . We first see the values that  $L_{DR}$  take for different values of  $yf(\mathbf{x})$ . Table 8 shows how  $L_{DR}$  changes as a function of  $yf(\mathbf{x})$ .

Interval	$L_{DR}$
$yf(\mathbf{x}) \in (\rho + \mu, \infty)$	0
$yf(\mathbf{x}) \in [\rho - \mu^2, \rho + \mu]$	$\frac{d}{\mu}(\mu - yf(\mathbf{x}) + \rho)$
$yf(\mathbf{x}) \in (-\rho + \mu, \rho - \mu^2)$	$(1 + \mu)d$
$yf(\mathbf{x}) \in [-\rho - \mu^2, -\rho + \mu]$	$(1 + \mu)d + \frac{(1-d)}{\mu}(\mu - yf(\mathbf{x}) - \rho)$
$yf(\mathbf{x}) \in (-\infty, -\rho - \mu^2)$	$1 + \mu$

**Table 8.**  $L_{DR}$  in different intervals (Proof for Theorem 1.(iii))

Now we take the limit  $\mu \rightarrow 0$ , which is shown in Table 9. We see that  $\lim_{\mu \rightarrow 0} L_{DR} = L_{0-d-1}$ .

Interval	$\lim_{\mu \rightarrow 0} L_{DR}$	$L_{0-d-1}$
$yf(\mathbf{x}) \in (\rho, \infty)$	0	0
$yf(\mathbf{x}) = \rho$	$d$	$d$
$yf(\mathbf{x}) \in (-\rho, \rho)$	$d$	$d$
$yf(\mathbf{x}) = -\rho$	1	1
$yf(\mathbf{x}) \in (-\infty, -\rho)$	1	1

**Table 9.**  $\lim_{\mu \rightarrow 0} L_{DR}$  in different intervals (Proof for Theorem 1.(iii))

3. In the rejection region  $yf(\mathbf{x}) \in (\rho - \mu^2, -\rho + \mu)$ , the loss remains constant, that is  $L_{DR}(f(\mathbf{x}), \rho, y) = d(1 + \mu)$ . This can be seen in Table 8.
4. For  $\mu > 0$ ,  $L_{DR} \leq (1 + \mu)$ ,  $\forall \rho \geq 0, \forall d \geq 0$ . This can be seen in Table 8.

5. When  $\rho = 0$ ,  $L_{\text{DR}}$  becomes

$$\begin{aligned} L_{\text{DR}}(f(\mathbf{x}), 0, y) &= \frac{d}{\mu} \left[ [\mu - yf(\mathbf{x})]_+ - [-\mu^2 - yf(\mathbf{x})]_+ \right] + \frac{(1-d)}{\mu} \left[ [\mu - yf(\mathbf{x}) - ]_+ \right. \\ &\quad \left. - [-\mu^2 - yf(\mathbf{x})]_+ \right] \\ &= \frac{1}{\mu} \left[ [\mu - yf(\mathbf{x})]_+ - [-\mu^2 - yf(\mathbf{x})]_+ \right] \end{aligned}$$

which is same as the  $\mu$ -ramp loss function used for classification problems without rejection option.

6. We have to show that  $L_{\text{DR}}$  is non-convex function of  $(yf(\mathbf{x}), \rho)$ . From (iv), we know that  $L_{\text{DR}} \leq (1 + \mu)$ . That is,  $L_{\text{DR}}$  is bounded above. We show non-convexity of  $L_{\text{DR}}$  by contradiction.

Let  $L_{\text{DR}}$  be convex function of  $(yf(\mathbf{x}), \rho)$ . Let  $\mathbf{z} = (yf(\mathbf{x}), \rho)$ . We also rewrite  $L_{\text{DR}}(f(\mathbf{x}), \rho, y)$  as  $L_{\text{DR}}(\mathbf{z})$ . We choose two points  $\mathbf{z}_1, \mathbf{z}_2$  such that  $L_{\text{DR}}(\mathbf{z}_1) > L_{\text{DR}}(\mathbf{z}_2)$ . Thus, from the definition of convexity, we have

$$L_{\text{DR}}(\mathbf{z}_1) \leq \lambda L_{\text{DR}}\left(\frac{\mathbf{z}_1 - (1-\lambda)\mathbf{z}_2}{\lambda}\right) + (1-\lambda)L_{\text{DR}}(\mathbf{z}_2) \quad \forall \lambda \in (0, 1)$$

Hence,

$$\frac{L_{\text{DR}}(\mathbf{z}_1) - (1-\lambda)L_{\text{DR}}(\mathbf{z}_2)}{\lambda} \leq L_{\text{DR}}\left(\frac{\mathbf{z}_1 - (1-\lambda)\mathbf{z}_2}{\lambda}\right)$$

Now, since  $L_{\text{DR}}(\mathbf{z}_1) > L_{\text{DR}}(\mathbf{z}_2)$ ,

$$\frac{L_{\text{DR}}(\mathbf{z}_1) - (1-\lambda)L_{\text{DR}}(\mathbf{z}_2)}{\lambda} = \frac{L_{\text{DR}}(\mathbf{z}_1) - L_{\text{DR}}(\mathbf{z}_2)}{\lambda} + L_{\text{DR}}(\mathbf{z}_2) \rightarrow \infty \quad \text{as } \lambda \rightarrow 0^+$$

Thus  $\lim_{\lambda \rightarrow 0^+} L_{\text{DR}}\left(\frac{\mathbf{z}_1 - (1-\lambda)\mathbf{z}_2}{\lambda}\right) = \infty$ . But  $L_{\text{DR}}$  is upper bounded by  $(1+\mu)d$ . This contradicts that  $L_{\text{DR}}$  is convex.

## B Proof of Lemma 1

Let  $\Theta' = (\mathbf{w}', b', \rho')$  minimizes  $R(\Theta)$ , where  $\rho' < 0$ . Thus  $-\rho' > 0$ . Consider  $\Theta'' = (\mathbf{w}', b', -\rho')$  as another point.

$$\begin{aligned} R(\Theta') - R(\Theta'') &= \frac{C(1-2d)}{\mu} \sum_{n=1}^N \left\{ -[\mu - y_n f(\mathbf{x}_n) + \rho']_+ + [-\mu^2 - y_n f(\mathbf{x}_n) + \rho']_+ \right. \\ &\quad \left. + [\mu - y_n f(\mathbf{x}_n) - \rho']_+ - [-\mu^2 - y_n f(\mathbf{x}_n) - \rho']_+ \right\} \\ &= C(1-2d) \sum_{n=1}^N \left\{ L_{\text{ramp}}(y_n f(\mathbf{x}_n) + \rho') - L_{\text{ramp}}(y_n f(\mathbf{x}_n) - \rho') \right\} \end{aligned}$$

where  $L_{\text{ramp}}(t) = \frac{1}{\mu}([\mu - t]_+ - [-\mu^2 - t]_+)$  is a monotonically non-increasing function of  $t$  [12]. Since  $\rho' < 0$ , thus,  $y_n f(\mathbf{x}_n) + \rho' < y_n f(\mathbf{x}_n) - \rho'$ ,  $\forall n$ . This implies  $L_{\text{ramp}}(y_n f(\mathbf{x}_n) + \rho') \geq L_{\text{ramp}}(y_n f(\mathbf{x}_n) - \rho')$ ,  $\forall n$ . Also  $(1-2d) \geq 0$ , since  $0 \leq d \leq 0.5$ . Thus  $R(\Theta') - R(\Theta'') \geq 0$ , which contradicts that  $\Theta'$  minimizes  $R(\Theta)$ . Thus, at the minimum of  $R(\Theta)$ ,  $\rho$  must be non-negative.



## C Derivation of Dual Optimization Problem $\mathcal{D}^{(l+1)}$

$$\begin{aligned} \mathcal{P}^{(l+1)} : \quad & \min_{\mathbf{w}, b, \xi', \xi'', \rho} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N [d\xi'_n + (1-d)\xi''_n] + \sum_{n=1}^N \beta_n^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \rho] \\ & + \sum_{n=1}^N \beta_n^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] \\ \text{s.t.} \quad & y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq \rho + \mu - \xi'_n, \quad \xi'_n \geq 0, \quad n = 1 \dots N \\ & y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq -\rho + \mu - \xi''_n, \quad \xi''_n \geq 0, \quad n = 1 \dots N \end{aligned}$$

The Lagrangian for above problem will be:

$$\begin{aligned} \mathcal{L} = & \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{\mu} \sum_{n=1}^N [d\xi'_n + (1-d)\xi''_n] + \sum_{n=1}^N \beta_n^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \rho] + \\ & \sum_{n=1}^N \beta_n^{(l)} [y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] + \sum_{n=1}^N \alpha'_n [\rho + \mu - \xi'_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)] - \sum_{n=1}^N \eta'_n \xi'_n \\ & + \sum_{n=1}^N \alpha''_n [-\rho + \mu - \xi''_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)] - \sum_{n=1}^N \eta''_n \xi''_n \end{aligned}$$

where  $\alpha'_n$  is dual variable corresponding to constraint  $y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq \rho + \mu - \xi'_n$ ,  $\alpha''_n$  is dual variable corresponding to  $y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq -\rho + \mu - \xi''_n$ ,  $\eta'_n$  is dual variable corresponding to  $\xi'_n \geq 0$ ,  $\eta''_n$  is dual variable corresponding to  $\xi''_n \geq 0$ . We take the gradient of Lagrangian with respect to the primal variables. By equating the gradient to zero, we get the KKT conditions of optimality for this optimization problem.

$$\begin{cases} \mathbf{w} = \sum_{n=1}^N y_n [\alpha'_n + \alpha''_n - \beta_n^{(l)} - \beta_n^{(l)}] \phi(\mathbf{x}_n) \\ \sum_{n=1}^N y_n [\alpha'_n + \alpha''_n - \beta_n^{(l)} - \beta_n^{(l)}] \\ \eta'_n + \alpha'_n = \frac{Cd}{\mu} & n = 1 \dots N \\ \eta''_n + \alpha''_n = \frac{C(1-d)}{\mu} & n = 1 \dots N \\ \sum_{n=1}^N [\alpha'_n - \alpha''_n - \beta_n^{(l)} + \beta_n^{(l)}] = 0 \\ \eta'_n \xi'_n = 0, \quad \eta'_n \geq 0 & n = 1 \dots N \\ \eta''_n \xi''_n = 0, \quad \eta''_n \geq 0 & n = 1 \dots N \\ \alpha'_n [\mu - \xi'_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] = 0, \quad \alpha'_n \geq 0 & n = 1 \dots N \\ \alpha''_n [\mu - \xi''_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \rho] = 0, \quad \alpha''_n \geq 0 & n = 1 \dots N \end{cases}$$

We make the dual optimization problem simpler by changing the variables in following way:

$$\begin{cases} \gamma'_n = \alpha'_n - \beta_n^{(l)}, & n = 1 \dots N \\ \gamma''_n = \alpha''_n - \beta_n^{(l)}, & n = 1 \dots N \end{cases}$$

By changing these variables, the new KKT conditions in terms of  $\gamma'$  and  $\gamma''$  are

$$\begin{cases} \mathbf{w} = \sum_{n=1}^N y_n(\gamma'_n + \gamma''_n)\phi(\mathbf{x}_n) \\ \sum_{n=1}^N y_n(\gamma'_n + \gamma''_n) = 0 \\ \eta'_n + \gamma'_n + \beta_n^{(l)} = \frac{Cd}{\mu} & n = 1 \dots N \\ \eta''_n + \gamma''_n + \beta_n^{(l)} = \frac{C(1-d)}{\mu} & n = 1 \dots N \\ \sum_{n=1}^N (\gamma'_n - \gamma''_n) = 0 \\ \eta'_n \xi'_n = 0, \quad \eta'_n \geq 0 & n = 1 \dots N \\ \eta''_n \xi''_n = 0, \quad \eta''_n \geq 0 & n = 1 \dots N \\ (\gamma'_n + \beta_n^{(l)})[\mu - \xi'_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] = 0, \quad \gamma'_n + \beta_n^{(l)} \geq 0 & n = 1 \dots N \\ (\gamma''_n + \beta_n^{(l)})[\mu - \xi''_n - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) + \rho] = 0, \quad \gamma''_n + \beta_n^{(l)} \geq 0 & n = 1 \dots N \end{cases}$$

Using the KKT conditions in the Langrangian, we replace the primal variables  $(\mathbf{w}, b, \rho, \xi', \xi'')$  in terms of the dual variables  $(\gamma', \gamma'')$ . The dual optimization problem  $\mathcal{D}^{(l+1)}$  will become:

$$\begin{aligned} \mathcal{D}^{(l+1)} = \min_{\gamma', \gamma''} & \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N y_n y_m (\gamma'_n + \gamma''_n)(\gamma'_m + \gamma''_m) k(\mathbf{x}_n, \mathbf{x}_m) - \mu \sum_{n=1}^N (\gamma'_n + \gamma''_n) \\ \text{s.t.} & \begin{cases} -\beta_n^{(l)} \leq \gamma'_n \leq \frac{Cd}{\mu} - \beta_n^{(l)} & n = 1 \dots N \\ -\beta_n^{(l)} \leq \gamma''_n \leq \frac{C(1-d)}{\mu} - \beta_n^{(l)} & n = 1 \dots N \\ \sum_{n=1}^N y_n (\gamma'_n + \gamma''_n) = 0 \\ \sum_{n=1}^N (\gamma'_n - \gamma''_n) = 0 \end{cases} \end{aligned}$$

where  $\gamma' = [\gamma'_1 \ \gamma'_2 \dots \gamma'_N]^T$  and  $\gamma'' = [\gamma''_1 \ \gamma''_2 \dots \gamma''_N]^T$ .