Boosting with Abstention

Corinna Cortes
Google Research
New York, NY 10011
corinna@google.com

Giulia DeSalvo
Courant Institute
New York, NY 10012
desalvo@cims.nyu.edu

Mehryar Mohri
Courant Institute and Google
New York, NY 10012
mohri@cims.nyu.edu

Abstract

We present a new boosting algorithm for the key scenario of binary classification with abstention where the algorithm can abstain from predicting the label of a point, at the price of a fixed cost. At each round, our algorithm selects a pair of functions, a base predictor and a base abstention function. We define convex upper bounds for the natural loss function associated to this problem, which we prove to be calibrated with respect to the Bayes solution. Our algorithm benefits from general margin-based learning guarantees which we derive for ensembles of pairs of base predictor and abstention functions, in terms of the Rademacher complexities of the corresponding function classes. We give convergence guarantees for our algorithm along with a linear-time weak-learning algorithm for abstention stumps. We also report the results of several experiments suggesting that our algorithm provides a significant improvement in practice over two confidence-based algorithms.

1 Introduction

Classification with abstention is a key learning scenario where the algorithm can abstain from making a prediction, at the price of incurring a fixed cost. This is the natural scenario in a variety of common and important applications. An example is spoken-dialog applications where the system can redirect a call to an operator to avoid the cost of incorrectly assigning a category to a spoken utterance and misguiding the dialog manager. This requires the availability of an operator, which incurs a fixed and predefined price. Other examples arise in the design of a search engine or an information extraction system, where, rather than taking the risk of displaying an irrelevant document, the system can resort to the help of a more sophisticated, but more time-consuming classifier. More generally, this learning scenario arises in a wide range of applications including health, bioinformatics, astronomical event detection, active learning, and many others, where abstention is an acceptable option with some cost. Classification with abstention is thus a highly relevant problem.

The standard approach for tackling this problem is via confidence-based abstention: a real-valued function h is learned for the classification problem and the points x for which its magnitude $|h(x)|$ is smaller than some threshold γ are rejected. Bartlett and Wegkamp [1] gave a theoretical analysis of this approach based on consistency. They introduced a discontinuous loss function taking into account the cost for rejection, upper-bounded that loss by a convex and continuous Double Hinge Loss (DHL) surrogate, and derived an algorithm based on that convex surrogate loss. Their work inspired a series of follow-up papers that developed both the theory and practice behind confidence-based abstention [32, 15, 31]. Further related works can be found in Appendix A.

In this paper, we present a solution to the problem of classification with abstention that radically departs from the confidence-based approach. We introduce a general model where a pair (h, r) for a classifier h and rejection function r are learned simultaneously. Under this novel framework, we present a Boosting-style algorithm with Abstention, BA, that learns accurately the classifier and abstention functions. Note that the terminology of "boosting with abstention" was used by Schapire and Singer [26] to refer to a scenario where a base classifier is allowed to abstain, but

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

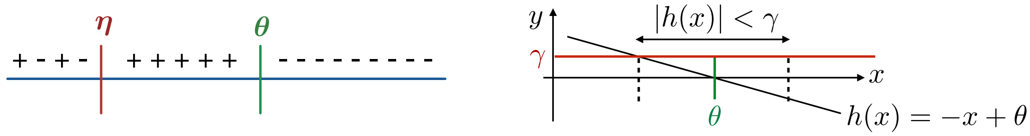

Figure 1: The best predictor h is defined by the threshold θ: $h(x) = -x + \theta$. For $c < \frac{1}{2}$, the region defined by $X \leq \eta$ should be rejected. But the corresponding abstention function r defined by $r(x) = x - \eta$ cannot be defined as $|h(x)| \leq \gamma$ for any $\gamma > 0$.

where the boosting algorithm itself has to commit to a prediction. This is therefore distinct from the scenario of classification with abstention studied here. Nevertheless, we will introduce and examine a confidence-based Two-Step Boosting algorithm, the TSB algorithm, that consists of first training Adaboost and next searching for the best confidence-based abstention threshold.

The paper is organized as follows. Section 2 describes our general abstention model which consists of learning a pair (h, r) simultaneously and compares it with confidence-based models. Section 3.2 presents a series of theoretical results for the problem of learning convex ensembles for classification with abstention, including the introduction of calibrated convex surrogate losses and general data-dependent learning guarantees. In Section 4, we use these learning bounds to design a regularized boosting algorithm. We further prove the convergence of the algorithm and present a linear-time weak-learning algorithm for a natural family of *abstention stumps*. Finally, in Section 5, we report several experimental results comparing the BA algorithm with the DHL and the TSB algorithms.

2 Preliminaries

In this section, we first introduce a general model for learning with abstention [7] and then compare it with confidence-based models.

2.1 General abstention model

We assume as in standard supervised learning that the training and test points are drawn i.i.d. according to some fixed but unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$. We consider the learning scenario of binary classification with abstention. Given an instance $x \in \mathcal{X}$, the learner has the option of abstaining from making a prediction for x at the price of incurring a non-negative loss $c(x)$, or otherwise making a prediction $h(x)$ using a predictor h and incurring the standard zero-one loss $1_{yh(x) \leq 0}$ where the true label is y. Since a random guess achieves an expected cost of at most $\frac{1}{2}$, rejection only makes sense for $c(x) < \frac{1}{2}$.

We will model the learner by a pair (h, r) where the function $r \colon \mathcal{X} \to \mathbb{R}$ determines the points $x \in \mathcal{X}$ to be rejected according to $r(x) \leq 0$ and where the hypothesis $h \colon \mathcal{X} \to \mathbb{R}$ predicts labels for non-rejected points via its sign. Extending the loss function considered in Bartlett and Wegkamp [1], the abstention loss for a pair (h, r) is defined as as follows for any $(x, y) \in \mathcal{X} \times \{-1, +1\}$:

$$L(h, r, x, y) = 1_{yh(x) \leq 0} 1_{r(x) > 0} + c(x) 1_{r(x) \leq 0}. \tag{1}$$

The abstention cost $c(x)$ is assumed known to the learner. In the following, we assume that c is a constant function, but part of our analysis is applicable to the more general case.

We denote by $\mathcal{H}$ and $\mathcal{R}$ two families of functions mapping $\mathcal{X}$ to $\mathbb{R}$ and we assume the labeled sample $S = ((x_1, y_1), \ldots, (x_m, y_m))$ is drawn i.i.d. from $\mathcal{D}^m$. The learning problem consists of determining a pair $(h, r) \in \mathcal{H} \times \mathcal{R}$ that admits a small expected abstention loss $R(h, r)$, defined as follows:

$$R(h, r) = \mathop{\mathbb{E}}_{(x,y) \sim \mathcal{D}} \left[1_{yh(x) \leq 0} 1_{r(x) > 0} + c 1_{r(x) \leq 0} \right]. \tag{2}$$

Similarly, we define the empirical loss of a pair $(h, r) \in \mathcal{H} \times \mathcal{R}$ over the sample S by: $\widehat{R}_S(h, r) = \mathbb{E}_{(x,y) \sim S} \left[1_{yh(x) \leq 0} 1_{r(x) > 0} + c 1_{r(x) \leq 0} \right]$, where $(x, y) \sim S$ indicates that (x, y) is drawn according to the empirical distribution defined by S.

2.2 Confidence-based abstention model

Confidence-based models are a special case of the general model for learning with rejection presented in Section 2.1 corresponding to the pair $(h(x), r(x)) = (h(x), |h(x)| - \gamma)$, where γ is a parameter that changes the threshold of rejection. This specific choice was based on consistency results shown in [1]. In particular, the Bayes solution (h^*, r^*) of the learning problem, that is where the distribution $\mathcal{D}$ is known, is given by $h^*(x) = \eta(x) - \frac{1}{2}$ and $r^*(x) = |h^*(x)| - (\frac{1}{2} - c)$ where $\eta(x) = \mathbb{P}[Y = +1|x]$ for any $x \in \mathcal{X}$, but note that this is not a unique solution. The form of $h^*(x)$ follows by a similar reasoning as for the standard binary classification problem. It is straightforward to see that the optimal rejection function r^* is non-positive, meaning a point is rejected, if and only if $\min\{\eta(x), 1 - \eta(x)\} \geq c$. Equivalently, the following holds: $\max\{\eta(x) - \frac{1}{2}, \frac{1}{2} - \eta(x)\} \leq \frac{1}{2} - c$ if and only if $|\eta(x) - \frac{1}{2}| \leq \frac{1}{2} - c$ and using the definition of h^*, we recover the optimal r^*. In light of the Bayes solution, the specific choice of the abstention function r is natural; however, requiring the abstention function r to be defined as $r(x) = |h(x)| - \gamma$, for some $h \in \mathcal{H}$, is in general too restrictive when predictors are selected out of a limited subset $\mathcal{H}$ of all measurable functions over $\mathcal{X}$. Consider the example shown in Figure 1 where $\mathcal{H}$ is a family of linear functions. For this simple case, the optimal abstention region cannot be attained as a function of the best predictor h while it can be achieved by allowing to learn a pair (h, r). Thus, the general model for learning with abstention analyzed in Section 2.1 is both more flexible and more general.

3 Theoretical analysis

This section presents a theoretical analysis of the problem of learning convex ensembles for classification with abstention. We first introduce general convex surrogate functions for the abstention loss and prove a necessary and sufficient condition based on their parameters for them to be calibrated. Next we define the ensemble family we consider and prove general data-dependent learning guarantees for it based on the Rademacher complexities of the base predictor and base rejector sets.

3.1 Convex surrogates

We introduce two types of convex surrogate functions for the abstention loss. Observe that the abstention loss $L(h, r, x, y)$ can be equivalently expressed as $L(h, r, x, y) = \max\left(1_{yh(x)\leq 0} 1_{-r(x)<0}, c\, 1_{r(x)\leq 0}\right)$. In view of that, since for any $f, g \in \mathbb{R}$, $\max(f, g) = \frac{f+g+|g-f|}{2} \geq \frac{f+g}{2}$, the following inequalities hold for $a > 0$ and $b > 0$:

$$\begin{aligned} L(h, r, x, y) &= \max\left(1_{yh(x)\leq 0} 1_{-r(x)<0}, c\, 1_{r(x)\leq 0}\right) \\ &\leq \max\left(1_{\max(yh(x), -r(x))\leq 0}, c\, 1_{r(x)\leq 0}\right) \\ &\leq \max\left(1_{\frac{yh(x)-r(x)}{2}\leq 0}, c\, 1_{r(x)\leq 0}\right) \\ &= \max\left(1_{a\,[yh(x)-r(x)]\leq 0}, c 1_{b\, r(x)\leq 0}\right) \\ &\leq \max\Big(\Phi_1\big(a\,[r(x) - yh(x)]\big), c\,\Phi_2\big(-b\,r(x)\big)\Big), \end{aligned}$$

where $u \to \Phi_1(-u)$ and $u \to \Phi_2(-u)$ are two non-increasing convex functions upper-bounding $u \to 1_{u\leq 0}$ over $\mathbb{R}$. Let L_{MB} be the convex surrogate defined by the last inequality above:

$$L_{\mathrm{MB}}(h, r, x, y) = \max\Big(\Phi_1\big(a\,[r(x) - yh(x)]\big), c\,\Phi_2\big(-b\,r(x)\big)\Big), \tag{3}$$

Since L_{MB} is not differentiable everywhere, we upper-bound the convex surrogate L_{MB} as follows: $\max\left(1_{a\,[yh(x)-r(x)]\leq 0}, c\, 1_{b\, r(x)\leq 0}\right) \leq \Phi_1\big(a\,[r(x) - yh(x)]\big) + c\,\Phi_2\big(-b\,r(x)\big)$. Similarly, we let L_{SB} denote this convex surrogate:

$$L_{\mathrm{SB}}(h, r, x, y) = \Phi_1\big(a\,[r(x) - yh(x)]\big) + c\,\Phi_2\big(-b\,r(x)\big). \tag{4}$$

Figure 2 shows the plots of the convex surrogates L_{MB} and L_{SB} as well as that of the abstention loss.

Let (h_L^*, r_L^*) denote the pair that attains the minimum of the expected loss $\mathbb{E}_{x,y}(L_{\mathrm{SB}}(h, r, x, y))$ over all measurable functions for $\Phi_1(u) = \Phi_2(u) = \exp(u)$. In Appendix F, we show that with $\eta(x) = \mathbb{P}(Y = +1|X = x)$, the pair (h_L^*, r_L^*) where $h_L^* = \frac{1}{2a}\log\left(\frac{\eta}{1-\eta}\right)$ and $r_L^* = \frac{1}{a+b}\log\left(\frac{cb}{2a}\sqrt{\frac{1}{\eta(1-\eta)}}\right)$ makes L_{SB} a calibrated loss, meaning that the sign of the (h_L^*, r_L^*) that minimizes the expected surrogate loss matches the sign of the Bayes classifier (h^*, r^*). More precisely, the following holds.

Theorem 1 (Calibration of convex surrogate)**.** *For $a > 0$ and $b > 0$, the* $\inf_{(h,r)} \mathbb{E}_{(x,y)}[L(h, r, x, y)]$ *is attained at (h_L^*, r_L^*) such that* $\mathrm{sign}(h^*) = \mathrm{sign}(h_L^*)$ *and* $\mathrm{sign}(r^*) = \mathrm{sign}(r_L^*)$ *if and only if* $b/a = 2\sqrt{(1-c)/c}$.

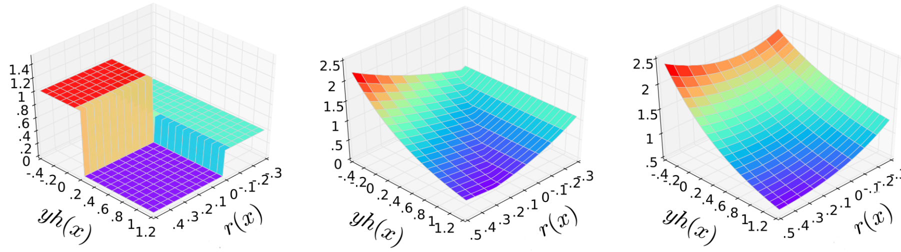

Figure 2: The left figure is a plot of the abstention loss. The middle figure is a plot of the surrogate function L_{MB} while the right figure is a plot of the surrogate loss L_{SB} both for $c = 0.45$.

The theorem shows that the classification and rejection solution obtained by minimizing the surrogate loss for that choice of (a, b) coincides with the one obtained using the original loss. In the following, we make the explicit choice of $a = 1$ and $b = 2\sqrt{(1-c)/c}$ for the loss L_{SB} to be calibrated.

3.2 Learning guarantees for ensembles in classification with abstention

In the standard scenario of classification, it is often easy to come up with simple base classifiers that may abstain. As an example, a simple rule could classify a message as spam based on the presence of some word, as ham in the presence of some other word, and just abstain in the absence of both, as in the boosting with abstention algorithm by Schapire and Singer [26]. Our objective is to learn ensembles of such base hypotheses to create accurate solutions for classification with abstention. Our ensemble functions are based on the framework described in Section 2.1. Let $\mathcal{H}$ and $\mathcal{R}$ be two families of functions mapping $\mathcal{X}$ to $[-1, 1]$. The ensemble family $\mathcal{F}$ that we consider is then the convex hull of $\mathcal{H} \times \mathcal{R}$:

$$\mathcal{F} = \left\{ \Big(\sum_{t=1}^{T} \alpha_t h_t, \sum_{t=1}^{T} \alpha_t r_t \Big) : T \geq 1, \alpha_t \geq 0, \sum_{t=1}^{T} \alpha_t = 1, h_t \in \mathcal{H}, r_t \in \mathcal{R} \right\}. \tag{5}$$

Thus, $(\mathbf{h}, \mathbf{r}) \in \mathcal{F}$ abstains on input $x \in \mathcal{X}$ when $\mathbf{r}(x) \leq 0$ and predicts the label $\operatorname{sign}(\mathbf{h}(x))$ otherwise.

Let $u \to \Phi_1(-u)$ and $u \to \Phi_2(-u)$ be two strictly decreasing differentiable convex function upper-bounding $u \to 1_{u \leq 0}$ over $\mathbb{R}$. For calibration constants $a, b > 0$, and cost $c > 0$, we assume that there exist u and v such that $\Phi_1(a\,u) < 1$ and $c\,\Phi_2(v) < 1$, otherwise the surrogate would not be useful. Let Φ_1^{-1} and Φ_2^{-1} be the inverse functions, which always exist since Φ_1 and Φ_2 are strictly monotone. We will use the following definitions: $C_{\Phi_1} = 2a\,\Phi_1'\big(\Phi_1^{-1}(1)\big)$ and $C_{\Phi_2} = 2cb\,\Phi_2'\big(\Phi_2^{-1}(1/c)\big)$. Observe that for $\Phi_1(u) = \Phi_2(u) = \exp(u)$, we simply have $C_{\Phi_1} = 2a$ and $C_{\Phi_2} = 2b$.

Theorem 2. *Let $\mathcal{H}$ and $\mathcal{R}$ be two families of functions mapping $\mathcal{X}$ to $\mathbb{R}$. Assume $N > 1$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S of size m from $\mathcal{D}$, the following holds for all $(\mathbf{h}, \mathbf{r}) \in \mathcal{F}$:*

$$R(\mathbf{h}, \mathbf{r}) \leq \mathop{\mathbb{E}}_{(x,y)\sim S}[L_{\text{MB}}(\mathbf{h}, \mathbf{r}, x, y)] + C_{\Phi_1} \mathfrak{R}_m(\mathcal{H}) + (C_{\Phi_1} + C_{\Phi_2}) \mathfrak{R}_m(\mathcal{R}) + \sqrt{\frac{\log 1/\delta}{2m}}.$$

The proof is given in Appendix C. The theorem gives effective learning guarantees for ensemble pairs $(\mathbf{h}, \mathbf{r}) \in \mathcal{F}$ when the base predictor and abstention functions admit favorable Rademacher complexities. In earlier work [7], we present a learning bound for a different type of surrogate losses which can also be extended to hold for ensembles.

Next, we derive margin-based guarantees in the case where $\Phi_1(u) = \Phi_2(u) = \exp(u)$. For any $\rho > 0$, the margin-losses associated to L_{MB} and L_{SB} are denoted by L^{ρ}_{MB} and L^{ρ}_{SB} and defined for all $(\mathbf{h}, \mathbf{r}) \in \mathcal{F}$ and $(x, y) \in \mathcal{X} \times \{-1, +1\}$ by

$$L^{\rho}_{\text{MB}}(\mathbf{h}, \mathbf{r}, x, y) = L_{\text{MB}}(\mathbf{h}/\rho, \mathbf{r}/\rho, x, y) \quad \text{and} \quad L^{\rho}_{\text{SB}}(\mathbf{h}, \mathbf{r}, x, y) = L_{\text{SB}}(\mathbf{h}/\rho, \mathbf{r}/\rho, x, y).$$

Theorem 2 applied to this margin-based loss results in the following corollary.

Corollary 3. *Assume $N > 1$ and fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of an i.i.d. sample S of size m from $\mathcal{D}$, the following holds for all $\mathbf{f} \in \mathcal{F}$:*

$$R(\mathbf{h}, \mathbf{r}) \leq \mathop{\mathbb{E}}_{(x,y)\sim S}[L^{\rho}_{\text{MB}}(\mathbf{h}, \mathbf{r}, x, y)] + \frac{2a}{\rho} \mathfrak{R}_m(\mathcal{H}) + \frac{2(a+b)}{\rho} \mathfrak{R}_m(\mathcal{R}) + \sqrt{\frac{\log 1/\delta}{2m}}.$$

BA($S = ((x_1, y_1), \ldots, (x_m, y_m))$)
for $i \leftarrow 1$ **to** m **do**
    $D_1(i, 1) \leftarrow \frac{1}{2m}$; $D_1(i, 2) \leftarrow \frac{1}{2m}$
for $t \leftarrow 1$ **to** T **do**
    $Z_{1,t} \leftarrow \sum_{i=1}^m D_t(i, 1)$; $Z_{2,t} \leftarrow \sum_{i=1}^m D_t(i, 2)$
    $k \leftarrow \operatorname{argmin}_{j \in [1,N]} 2Z_{1,t}\epsilon_{t,j} + Z_{1,t}\overline{r}_{j,1} - 2\sqrt{c(1-c)} Z_{2,t}\overline{r}_{j,2}$ ▷ Direction
    $Z \leftarrow Z_{1,t}(\epsilon_{t,k} + \frac{\overline{r}_{k,1}}{2}) - 2\sqrt{c(1-c)} Z_{2,t}\frac{\overline{r}_{k,2}}{2}$
    if $(Z_{1,t} - Z)e^{\alpha_{t-1,k}} - Ze^{-\alpha_{t-1,k}} < \frac{m}{Z_t}\beta$ **then**
        $\eta_t \leftarrow -\alpha_{t-1,k}$ ▷ Step
    else $\eta_t \leftarrow \log\left[-\frac{m\beta}{2Z_t Z} + \sqrt{\left[\frac{m\beta}{2Z_t Z}\right]^2 + \frac{Z_{1,t}}{Z} - 1}\right]$ ▷ Step
    $\boldsymbol{\alpha}_t \leftarrow \boldsymbol{\alpha}_{t-1} + \eta_t \mathbf{e}_k$
    $\mathbf{r}_t \leftarrow \sum_{j=1}^N \alpha_j r_j$
    $\mathbf{h}_t \leftarrow \sum_{j=1}^N \alpha_j h_j$
    $Z_{t+1} \leftarrow \sum_{i=1}^m \Phi'\big(\mathbf{r}_t(x_i) - y_i \mathbf{h}_t(x_i)\big) + \Phi'\big(-2\sqrt{\frac{1-c}{c}}\mathbf{r}_t(x_i)\big)$
    for $i \leftarrow 1$ **to** m **do**
        $D_{t+1}(i, 1) \leftarrow \frac{\Phi'\left(\mathbf{r}_t(x_i) - y_i \mathbf{h}_t(x_i)\right)}{Z_{t+1}}$; $D_{t+1}(i, 2) \leftarrow \frac{\Phi'\left(-2\sqrt{\frac{1-c}{c}}\mathbf{r}_t(x_i)\right)}{Z_{t+1}}$
$(\mathbf{h}, \mathbf{r}) \leftarrow \sum_{j=1}^N \alpha_{T,j}(\mathbf{h}_j, \mathbf{r}_j)$
return $(\mathbf{h}, \mathbf{r})$

Figure 3: Pseudocode of the BA algorithm for both the exponential loss with $\Phi_1(u) = \Phi_2(u) = \exp(u)$ as well as for the logistic loss with $\Phi_1(u) = \Phi_2(u) = \log_2(1 + e^u)$. The parameters include the cost of rejection c and β determining the strength of the the $\boldsymbol{\alpha}$-constraint for the L1 regularization. The definition of the weighted errors $\epsilon_{t,k}$ as well as the expected rejections, $\overline{r}_{k,1}$ and $\overline{r}_{k,2}$, are given in Equation 7. For other surrogate losses, the step size η_t is found via a line search or other numerical methods by solving $\operatorname{argmin}_\eta F(\boldsymbol{\alpha}_{t-1} + \eta \mathbf{e}_k)$.

The bound of Corollary 3 applies similarly to L_{SB}^ρ since it is an upper bound on L_{MB}^ρ. It can further be shown to hold uniformly for all $\rho \in (0, 1)$ at the price of a term in $O\left(\sqrt{\frac{\log \log 1/\rho}{m}}\right)$ using standard techniques [16, 22] (see Appendix C).

4 Boosting algorithm

Here, we derive a boosting-style algorithm (BA algorithm) for learning an ensemble with the option of abstention for both losses L_{MB} and L_{SB}. Below, we describe the algorithm for L_{SB} and refer the reader to Appendix H for the version using the loss L_{MB}.

4.1 Objective function

The BA algorithm solves a convex optimization problem that is based on Corollary 3 for loss L_{SB}. Since the last three terms of the right-hand side of the bound of the corollary do not depend on $\boldsymbol{\alpha}$, this suggests to select $\boldsymbol{\alpha}$ as the solution of $\min_{\boldsymbol{\alpha} \in \Delta} \frac{1}{m} \sum_{i=1}^m L_{\text{SB}}^\rho(\mathbf{h}, \mathbf{r}, x_i, y_i)$. Via a change of variable $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}/\rho$ that does not affect the optimization problem, we can equivalently search for $\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m} \sum_{i=1}^m L_{\text{SB}}(\mathbf{h}, \mathbf{r}, x_i, y_i)$ such that $\sum_{t=1}^T \alpha_t \leq 1/\rho$. Introducing the Lagrange variable β associated to the constraint $\sum_{t=1}^T \alpha_t \leq 1/\rho$, the problem can rewritten as: $\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{m} \sum_{i=1}^m L_{\text{SB}}(\mathbf{h}, \mathbf{r}, x_i, y_i) + \beta \sum_{t=1}^T \alpha_t$. Letting $\{(h_1, r_1), \ldots, (h_N, r_N)\}$ be the set of base functions pairs for the classifier and rejection function, we can rewrite the optimization problem as the minimization over $\boldsymbol{\alpha} \geq 0$ of

$$\frac{1}{m}\sum_{i=1}^{m}\Phi\Big(\sum_{j=1}^{N}\alpha_j r_j(x_i) - y_i\sum_{j=1}^{N}\alpha_j h_j(x_i)\Big) + c\,\Phi\Big(-b\sum_{j=1}^{N}\alpha_j r_j(x_i)\Big) + \beta\sum_{j=1}^{N}\alpha_j.$$

Thus, the following is the objective function of our optimization problem:

$$F(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^{m}\Phi\big(\mathbf{r}_t(x_i) - y_i\mathbf{h}_t(x_i)\big) + c\,\Phi\big(-b\,\mathbf{r}_t(x_i)\big) + \beta\sum_{j=1}^{N}\alpha_j. \tag{6}$$

4.2 Projected coordinate descent

The problem $\min_{\boldsymbol{\alpha}\geq 0} F(\boldsymbol{\alpha})$ is a convex optimization problem, which we solve via projected coordinate descent. Let $\mathbf{e}_k$ be the kth unit vector in $\mathbb{R}^N$ and let $F'(\boldsymbol{\alpha}, \mathbf{e}_j)$ be the directional derivative of F along the direction $\mathbf{e}_j$ at $\boldsymbol{\alpha}$. The algorithm consists of the following three steps. First, it determines the direction of maximal descent by $k = \operatorname{argmax}_{j\in[1,N]} |F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j)|$. Second, it calculates the best step η along the direction that preserves non-negativity of $\boldsymbol{\alpha}$ by $\eta = \operatorname{argmin}_{\boldsymbol{\alpha}_{t-1}+\eta\mathbf{e}_k \geq 0} F(\boldsymbol{\alpha}_{t-1} + \eta\mathbf{e}_k)$. Third, it updates $\boldsymbol{\alpha}_{t-1}$ to $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + \eta\mathbf{e}_k$.

The pseudocode of the BA algorithm is given in Figure 3. The step and direction are based on $F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j)$. For any $t \in [1, T]$, define a distribution D_t over the pairs (i, n), with n in $\{1, 2\}$

$$D_t(i,1) = \frac{\Phi'\big(\mathbf{r}_{t-1}(x_i) - y_i\mathbf{h}_{t-1}(x_i)\big)}{Z_t} \quad \text{and} \quad D_t(i,2) = \frac{\Phi'\big(-b\,\mathbf{r}_{t-1}(x_i)\big)}{Z_t},$$

where Z_t is the normalization factor given by $Z_t = \sum_{i=1}^m \Phi'\big(\mathbf{r}_{t-1}(x_i) - y_i\mathbf{h}_{t-1}(x_i)\big) + \Phi'\big(-b\,\mathbf{r}_{t-1}(x_i)\big)$. In order to derive an explicit formulation of the descent direction that is based on the weighted error of the classification function h_j and the expected value of the rejection function r_j, we use the distributions $D_{1,t}$ and $D_{2,t}$ defined by $D_t(i,1)/Z_{1,t}$ and $D_t(i,1)/Z_{2,t}$ where $Z_{1,t} = \sum_{i=1}^m D_t(i,1)$ and $Z_{2,t} = \sum_{i=1}^m D_t(i,2)$ are the normalization factors. Now, for any $j \in [1, N]$ and $s \in [1, T]$, we can define the weighted error $\epsilon_{t,j}$ and the expected value of the rejection function, $\overline{r}_{j,1}$ and $\overline{r}_{j,2}$, over distribution $D_{1,t}$ and $D_{2,t}$ as follows:

$$\epsilon_{t,j} = \tfrac{1}{2}\Big[1 - \mathop{\mathbb{E}}_{i\sim D_{1,t}}[y_i h_j(x_i)]\Big], \quad \overline{r}_{j,1} = \mathop{\mathbb{E}}_{i\sim D_{1,t}}[r_j(x_i)], \text{ and } \overline{r}_{j,2} = \mathop{\mathbb{E}}_{i\sim D_{2,t}}[r_j(x_i)]. \tag{7}$$

Using these definition, we show (see Appendix D) that the descent direction is given by

$$k = \operatorname*{argmin}_{j\in[1,N]} 2Z_{1,t}\epsilon_{t,j} + Z_{1,t}\overline{r}_{j,1} - 2\sqrt{c(1-c)}Z_{2,t}\overline{r}_{j,2}.$$

This equation shows that $Z_{1,t}$ and $2\sqrt{c(1-c)}Z_{2,t}$ re-scale the weighted error and expected rejection. Thus, finding the best descent direction by minimizing this equation is equivalent to finding the best scaled trade-off between the misclassification error and the average rejection cost. The step size can in general be found via line search or other numerical methods, but we have derived a closed-form solution of the step size for both the exponential and logistic loss (see Appendix D.2). Further details of the derivation of projected coordinate descent on F are also given in Appendix D.

Note that for $\mathbf{r}_t \to 0^+$ in Equation 6, that is when the rejection terms are dropped in the objective, we retrieve the L1-regularized Adaboost. As for Adaboost, we can define a *weak learning assumption* which requires that the directional derivative along at least one base pair be non-zero. For $\beta = 0$, it does not hold when for all j: $2\epsilon_{s,j} - 1 = -\overline{r}_{j,1} + \frac{2\sqrt{c(1-c)}Z_{2,t}}{Z_{1,t}}\overline{r}_{j,2}$, which corresponds to a balance between the edge and rejection costs for all j. Observe that in the particular case when the rejection functions are zero, it coincides with the standard weak learning assumption for Adaboost ($\epsilon_{s,j} = \frac{1}{2}$ for all j).

The following theorem provides the convergence of the projected coordinate descent algorithm for our objective function, $F(\boldsymbol{\alpha})$. The proof is given in Appendix E.

Theorem 4. *Assume that Φ is twice differentiable and that $\Phi''(u) > 0$ for all $u \in \mathbb{R}$. Then, the projected coordinate descent algorithm applied to F converges to the solution $\boldsymbol{\alpha}^*$ of the optimization problem $\max_{\boldsymbol{\alpha}\geq 0} F(\boldsymbol{\alpha})$. If additionally Φ is strongly convex over the path of the iterates $\boldsymbol{\alpha}_t$ then there exists $\tau > 0$ and $\nu > 0$ such that for all $t > \tau$, $F(\boldsymbol{\alpha}_{t+1}) - F(\boldsymbol{\alpha}^*) \leq \big(1 - \frac{1}{\nu}\big)\big(F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*)\big)$.*

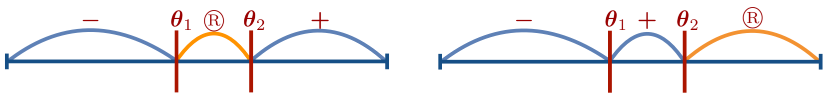

Figure 4: Illustration of the abstention stumps on a variable X.

Specifically, this theorem holds for the exponential loss $\Phi(u) = \exp(u)$ and the logistic loss $\Phi(-u) = \log_2(1 + e^{-u})$ since they are strongly convex over the compact set containing the $\boldsymbol{\alpha}_t$s.

4.3 Abstention stumps

We first define a family of base hypotheses, *abstention stumps*, that can be viewed as extensions of the standard boosting stumps to the setting of classification with abstention. An abstention stump h_{θ_1,θ_2} over the feature X is defined by two thresholds $\theta_1, \theta_2 \in \mathbb{R}$ with $\theta_1 \leq \theta_2$. There are 6 different such stumps, Figure 4 illustrates two of them. For the left figure, points with variables X less than or equal to θ_1 are labeled negatively, those with $X \geq \theta_2$ are labeled positively, and those with X between θ_1 and θ_2 are rejected. In general, an abstention stump is defined by the pair $\big(h_{\theta_1,\theta_2}(X), r_{\theta_1,\theta_2}(X)\big)$ where, for Figure 4-left, $h_{\theta_1,\theta_2}(X) = -1_{X \leq \theta_1} + 1_{X > \theta_2}$ and $r_{\theta_1,\theta_2}(X) = 1_{\theta_1 < X \leq \theta_2}$.

Thus, our abstention stumps are pairs $(h, \hat{r})$ with h taking values in $\{-1, 0, 1\}$ and $\hat{r}$ in $\{0, 1\}$, and such that for any x either $h(x)$ or $\hat{r}(x)$ is zero. For our formulation and algorithm, these stumps can be used in combination with any $\gamma > 0$, to define a family of base predictor and base rejector pairs of the form $(h(x), \gamma - \hat{r}(x))$. Since α_t is non-negative, the value γ is needed to correct for over-rejection by previously selected abstention stumps. The γ can be automatically learned by adding to the set of base pairs the constant functions $(h_0, r_0) = (0, -1)$. An ensemble solution returned by the BA algorithm is therefore of the form $\big(\sum_t \alpha_t h_t(x), \sum_t \alpha_t r_t(x)\big)$ where α_ts are the weights assigned to each base pair.

Now, consider a sample of m points sorted by the value of X, which we denote by $X_1 \leq \cdots \leq X_m$. For abstention stumps, the derivative of the objective, F, can be further simplified (see Appendix G) such that the problem can be reduced to finding an abstention stump with the minimal expected abstention loss $l(\theta_1, \theta_2)$, that is

$$\operatorname*{argmin}_{\theta_1,\theta_2} \sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i \leq \theta_1} + 1_{y_i=-1}1_{X_i > \theta_2}] + \big(2D_t(i,1) - cb\, D_t(i,2)\big)1_{\theta_1 < X_i \leq \theta_2}.$$

Notice that given m points, at most $(m+1)$ thresholds need to be considered for θ_1 and θ_2. Hence, a straightforward algorithm inspects all possible $O(m^2)$ pairs (θ_1, θ_2) with $\theta_1 \leq \theta_2$ in time $O(m^2)$. However, Lemma 5 below and further derivations in Appendix G, allows for an $O(m)$-time algorithm for finding optimal abstention stumps when the problem is solved without the constraint $\theta_1 \leq \theta_2$. Note that while we state the lemma for the abstention stump in Figure 4-left, similar results hold for any of the 6 types of stumps.

Lemma 5. *The optimization problem without the constraint* $(\theta_1 < \theta_2)$ *can be decomposed as follows:*

$$\operatorname*{argmin}_{\theta_1,\theta_2} l(\theta_1, \theta_2) = \operatorname*{argmin}_{\theta_1} \sum_{i=1}^{m} 2D_t(i,1)1_{y_i=+1}1_{X_i \leq \theta_1} + \big(2D_t(i,1) - cb\, D_t(i,2)\big)1_{\theta_1 < X_i} \quad (8)$$

$$+ \operatorname*{argmin}_{\theta_2} \sum_{i=1}^{m} 2D_t(i,1)1_{y_i=-1}1_{X_i > \theta_2} + \big(2D_t(i,1) - cb\, D_t(i,2)\big)1_{X_i \leq \theta_2}. \quad (9)$$

The optimization Problems (8) and (9) can be solved in linear time, via a method similar to that of finding the optimal threshold for a standard zero-one loss boosting stump. When the condition $\theta_1 < \theta_2$ does not hold, we can simply revert to finding the minimum of $l(\theta_1, \theta_2)$ in the naive way. In practice, we find most often that the optimal solution of Problem 8 and Problem 9 satisfies $\theta_1 < \theta_2$.

5 Experiments

In this section, we present the results of experiments with our abstention stump BA algorithm based on L_{SB} for several datasets. We compare the BA algorithm with the DHL algorithm [1], as well as a

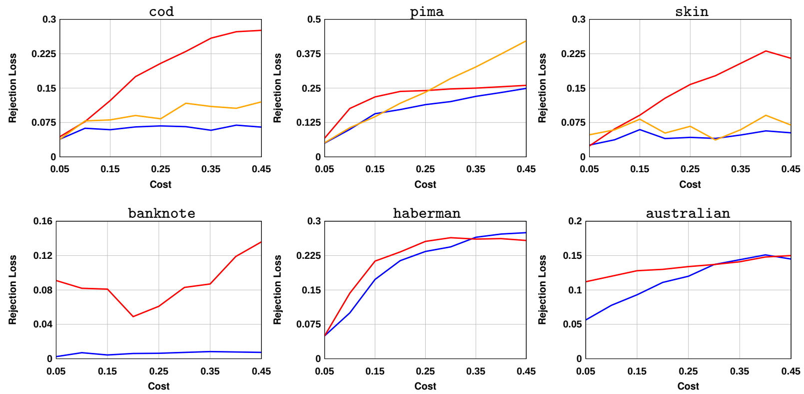

Figure 5: Average rejection loss on the test set as a function of the abstention cost c for the TSB Algorithm (in orange), the DHL Algorithm (in red) and the BA Algorithm (in blue) based on L_{SB}.

confidence-based boosting algorithm TSB. Both of these algorithms are described in further detail in Appendix B. We tested the algorithms on six data sets from UCI's data repository, specifically `australian`, `cod`, `skin`, `banknote`, `haberman`, and `pima`. For more information about the data sets, see Appendix I. For each data set, we implemented the standard 5-fold cross-validation where we randomly divided the data into training, validation and test set with the ratio 3:1:1. Using a different random partition, we repeated the experiments five times. For all three algorithms, the cost values ranged over $c \in \{0.05, 0.1, \ldots, 0.5\}$ while threshold γ ranged over $\gamma \in \{0.08, 0.16, \ldots, 0.96\}$. For the BA algorithm, the β regularization parameter ranged over $\beta \in \{0, 0.05, \ldots, 0.95\}$. All experiments for BA were based on $T = 200$ boosting rounds. The DHL algorithm used polynomial kernels with degree $d \in \{1, 2, 3\}$ and it was implemented in CVX [8]. For each cost c, the hyperparameter configuration was chosen to be the set of parameters that attained the smallest average rejection loss on the validation set. For that set of parameters we report the results on the test set.

We first compared the confidence-based TSB algorithm with the BA and DHL algorithms (first row of Figure 5). The experiments show that, while TSB can sometimes perform better than DHL, in a number of cases its performance is dramatically worse as a function of c and, in all cases it is outperformed by BA. In Appendix J, we give the full set of results for the TSB algorithm.

In view of that, our next series of results focus on the BA and DHL algorithms, directly designed to optimize the rejection loss, for 3 other datasets (second row of Figure 5). Overall, the figures show that BA outperforms the state-of-the-art DHL algorithm for most values of c, thereby indicating that BA yields a significant improvement in practice. We have also successfully run BA on the CIFAR-10 data set (boat and horse images) which contains 10,000 instances and we believe that our algorithm can scale to much larger datasets. In contrast, training DHL on such larger samples did not terminate as it is based on a costly QCQP. In Appendix J, we present tables that report the average and standard deviation of the abstention loss as well as the fraction of rejected points and the classification error on non-rejected points.

6 Conclusion

We introduced a general framework for classification with abstention where the predictor and abstention functions are learned simultaneously. We gave a detailed study of ensemble learning within this framework including: new surrogate loss functions proven to be calibrated, Rademacher complexity margin bounds for ensemble learning of the pair of predictor and abstention functions, a new boosting-style algorithm, the analysis of a natural family of base predictor and abstention functions, and the results of several experiments showing that BA algorithm yield a significant improvement over the confidence-based algorithms DHL and TSB. Our algorithm can be further extended by considering more complex base pairs such as more general ternary decision trees with rejection leaves. Moreover, our theory and algorithm can be generalized to the scenario of multi-class classification with abstention, which we have already initiated.

Acknowledgments

This work was partly funded by NSF CCF-1535987 and IIS-1618662.

References

[1] P. Bartlett and M. Wegkamp. Classification with a reject option using a hinge loss. *JMLR*, 2008.

[2] A. Bounsiar, E. Grall, and P. Beauseroy. Kernel based rejection method for supervised classification. In *WASET*, 2007.

[3] H. L. Capitaine and C. Frelicot. An optimum class-rejective decision rule and its evaluation. In *ICPR*, 2010.

[4] K. Chaudhuri and C. Zhang. Beyond disagreement-based agnostic active learning. In *NIPS*, 2014.

[5] C. Chow. An optimum character recognition system using decision function. *IEEE Trans. Comput.*, 1957.

[6] C. Chow. On optimum recognition error and reject trade-off. *IEEE Trans. Comput.*, 1970.

[7] C. Cortes, G. DeSalvo, and M. Mohri. Learning with rejection. In *ALT*, 2016.

[8] I. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0, Aug. 2012.

[9] B. Dubuisson and M. Masson. Statistical decision rule with incomplete knowledge about classes. In *PR*, 1993.

[10] R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *JMLR*, 2010.

[11] R. El-Yaniv and Y. Wiener. Agnostic selective classification. In *NIPS*, 2011.

[12] C. Elkan. The foundations of cost-sensitive learning. In *IJCAI*, 2001.

[13] G. Fumera and F. Roli. Support vector machines with embedded reject option. In *ICPR*, 2002.

[14] G. Fumera, F. Roli, and G. Giacinto. Multiple reject thresholds for improving classification reliability. In *ICAPR*, 2000.

[15] Y. Grandvalet, J. Keshet, A. Rakotomamonjy, and S. Canu. Suppport vector machines with a reject option. In *NIPS*, 2008.

[16] V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30, 2002.

[17] T. Landgrebe, D. Tax, P. Paclik, and R. Duin. The interaction between classification and reject performance for distance-based reject-option classifiers. *PRL*, 2005.

[18] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer, New York, 1991.

[19] M. Littman, L. Li, and T. Walsh. Knows what it knows: A framework for self-aware learning. In *ICML*, 2008.

[20] Z.-Q. Luo and P. Tseng. On the convergence of coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 1992.

[21] I. Melvin, J. Weston, C. S. Leslie, and W. S. Noble. Combining classifiers for improved classification of proteins from sequence or structure. *BMC Bioinformatics*, 2008.

[22] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. In *JMLR*, 2011.

[24] T. Pietraszek. Optimizing abstaining classifiers using ROC analysis. In *ICML*, 2005.

[25] C. Santos-Pereira and A. Pires. On optimal reject rules and ROC curves. *PRL*, 2005.

[26] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

[27] D. Tax and R. Duin. Growing a multi-class classifier with a reject option. In *Pattern Recognition Letters*, 2008.

[28] F. Tortorella. An optimal reject rule for binary classifiers. In *ICAPR*, 2001.

[29] K. Trapeznikov and V. Saligrama. Supervised sequential classification under budget constraints. In *AISTATS*, 2013.

[30] J. Wang, K. Trapeznikov, and V. Saligrama. An lp for sequential learning under budgets. In *JMLR*, 2014.

[31] M. Yuan and M. Wegkamp. Classification methods with reject option based on convex risk minimizations. In *JMLR*, 2010.

[32] M. Yuang and M. Wegkamp. Support vector machines with a reject option. In *Bernoulli*, 2011.

[33] C. Zhang and K. Chaudhuri. The extended littlestone's dimension for learning with mistakes and abstentions. In *COLT*, 2016.

A Extended Related Work

Initial work in learning with abstention has focused on the optimal trade-off between the error and abstention rate [5, 6] as well as finding the optimal abstention rule based on the ROC curve [14, 28, 25]. Another series of papers used rejection options to reduce misclassification rate, but theoretical learning guarantees were not given [13, 24, 2, 17, 21]. More recently, El-Yaniv and Wiener [10, 11] study the trade-off between the coverage and the accuracy of classifiers by using an approach related to active learning.

A seemingly connected framework is that of cost-sensitive learning where the cost of misclassifying class y_1 as class y_2 may depend on the pair (y_1, y_2) [12]. It would be tempting to view classification with abstention as a special instance of cost-sensitive learning with the set of classes $\{-1, +1, ®\}$, with ® standing for abstention and where a different cost would be assigned to abstention. However, in our problem, ® is not an intrinsic class: training or test samples bear no ® label. Instead, the distribution over that set will depend on the algorithm. Thus, classification with abstention cannot be cast as a special case of cost-sensitive learning. Sequential learning with a budget is also a marginally related task where abstention functions are learned. But, unlike our approach, it is done in a two-step process where the classifier function is fixed [29, 30]. Lastly, the option of abstaining has been analyzed in related topics including the multi-class setting [27, 9, 3], reinforcement learning [19], online learning [33] and active learning [4].

B Confidence-based abstention model

In this appendix, we describe two confidence-based abstention algorithms: the DHL algorithm and the TSB algorithm.

B.1 DHL algorithm

The DHL algorithm found in [1] is based on a double hinge loss, which is a hinge-type convex surrogate, with favorable consistency results. The optimization problem solved by the DHL algorithm minimizes this surrogate loss along with the constraint that the norm of the classifier is bounded by $1 - c$. More precisely, let $\mathcal{H}$ be a hypotheses sets defined in terms of PSD kernels K over $\mathcal{X}$ where $\mathbf{\Phi}$ is the feature mapping associated to K, then the DHL solves the following QCQP optimization problem

$$\begin{aligned}
\min_{\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta}} \; & \sum_{i=1}^{m} \xi_i + \frac{1-2c}{c} \beta_i \\
\text{subject to } & \sum_{i,j=1}^{m} \alpha_i \alpha_j K(x_i, x_j) \leq (1-c)^2 \\
& \xi_i \geq 1 - y_i \left(\sum_{i=1}^{m} \alpha_i K(x_i, x) \right) \wedge \xi_i \geq 0, \\
& \beta_i \geq -y_i \left(\sum_{i=1}^{m} \alpha_i K(x_i, x) \right) \wedge \beta_i \geq 0, i \in [1, m].
\end{aligned}$$

B.2 Two-step Adaboost (TSB)

The TSB algorithm is a confidence-based algorithm that proceeds in two steps. The first step consists of training a vanilla Adaboost algorithm which returns a classifier h. Then, given classifier h, the second step is to search for the best threshold γ that minimizes the empirical abstention loss. More precisely, we pick the parameter γ via cross-validation, by choosing the threshold that minimizes the empirical abstention loss on the validation set. This is a natural confidence-based boosting algorithm and since the BA algorithm is based on boosting, it provides a useful baseline for our experiments. We implemented this algorithm using scikit-learn [23].

C Theoretical guarantees

In this appendix, we provide the proof of the theoretical guarantees presented in Section 3.2.

Let $u \to \Phi_1(-u)$ and $u \to \Phi_2(-u)$ be two strictly non-increasing differentiable convex functions upper-bounding $u \to 1_{u \leq 0}$ over $\mathbb{R}$. We assume that $a\,, b\, > 0$, and $c > 0$. We will use the quantity $\min(\Phi_1(a\, u), 1)$ and so we assume that there exists u such that $\Phi_1(a\, u) < 1$ and similarly, we need to analyze $\min(c\Phi_2(u), 1)$ and so we assume there exists a u such that $c\Phi_2(u) < 1$. Note that if these two assumptions did not hold, then the surrogate would not be useful. Let Φ_1^{-1} and Φ_2^{-1} be the inverse functions, which always exist since Φ_1 and Φ_2 are strictly monotone functions. For simplicity, we define $C_{\Phi_1} = 2a\, \Phi_1'\big(\Phi_1^{-1}(1)\big)$ and $C_{\Phi_2} = 2cb\, \Phi_2'\big(\Phi_2^{-1}(1/c)\big)$.

Theorem 2. *Let $\mathcal{H}$ and $\mathcal{R}$ be family of functions mapping $\mathcal{X}$ to $\mathbb{R}$. Assume $N > 1$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S of size m from $\mathcal{D}$, the following holds for all $(\mathbf{h}, \mathbf{r}) \in \mathcal{F}$:*

$$R(\mathbf{h}, \mathbf{r}) \leq \mathop{\mathbb{E}}_{(x,y)\sim S}[L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y)] + C_{\Phi_1}\mathfrak{R}_m(\mathcal{H}) + (C_{\Phi_1} + C_{\Phi_2})\mathfrak{R}_m(\mathcal{R}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof. Let $\mathcal{L}_{\mathrm{MB},\mathcal{F}}$ be the family of functions defined by $\mathcal{L}_{\mathrm{MB},\mathcal{F}} = \big\{(x, y) \mapsto \min(L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y), 1), (\mathbf{h}, \mathbf{r}) \in \mathcal{F}\big\}$. Since $\min(L_{\mathrm{MB}}, 1)$ is bounded by one, by the general Rademacher complexity generalization bound [16], with probability at least $1 - \delta$ over the draw of a sample S, the following holds:

$$\begin{aligned}
R(\mathbf{h}, \mathbf{r}) &\leq \mathop{\mathbb{E}}_{(x,y)\sim \mathcal{D}}[\min(L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y), 1)] \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim S}[\min(L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y), 1)] + 2\mathfrak{R}_m(\mathcal{L}_{\mathrm{MB},\mathcal{F}}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}} \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim S}[L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y)] + 2\mathfrak{R}_m(\mathcal{L}_{\mathrm{MB},\mathcal{F}}) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.
\end{aligned}$$

Since for any $a, b \in \mathbb{R}$, $\min\big(\max(a, b), 1\big) = \max\big(\min(a, 1), \min(b, 1)\big)$, we can write

$$\begin{aligned}
&\min(L_{\mathrm{MB}}(\mathbf{h}, \mathbf{r}, x, y), 1) \\
&= \max\Big(\min\Big(\Phi_1\big(a\,[\mathbf{r}(x) - y\mathbf{h}(x)]\big), 1\Big), \min\Big(c\,\Phi_2\big(-b\,\mathbf{r}(x)\big), 1\Big)\Big) \\
&\leq \min\Big(\Phi_1\big(b\,[\mathbf{r}(x) - y\mathbf{h}(x)]\big), 1\Big) + \min\Big(c\,\Phi_2\big(-b\,\mathbf{r}(x)\big), 1\Big).
\end{aligned}$$

The function $\Phi_1\big(a\, u\big)$ has a non-negative increasing derivative because it is a strictly increasing convex function. Since $\min\big(\Phi_1\big(a\, u\big), 1\big) = \Phi_1\big(a\, u\big)$ for $a\, u \leq \Phi_1^{-1}(1)$, the Lipschitz constant of $u \mapsto \min\big(\Phi_1\big(a\, u\big), 1\big)$ is given by $a\, \Phi_1'(\Phi_1^{-1}(1))$. Similarly, $u \mapsto \min\big(c\Phi_2\big(b\, u\big), 1\big)$ is also $cb\, \Phi_2'(\Phi_2^{-1}(1/c))$-Lipschitz. Then, by Talagrand's lemma [18],

$$\begin{aligned}
\mathfrak{R}_m(\mathcal{L}_{\mathrm{MB},\mathcal{F}}) \leq{}& a\, \Phi_1'\big(\Phi_1^{-1}(1)\big)\mathfrak{R}_m\big((x, y) \mapsto \mathbf{r}(x) - y\mathbf{h}(x) \colon (\mathbf{h}, \mathbf{r}) \in \mathcal{F}\big) \\
&+ c\, b\ \Phi_2'\big(\Phi_2^{-1}(1/c)\big)\, \mathfrak{R}_m\big((x, y) \mapsto -\mathbf{r}(x) \colon (\mathbf{h}, \mathbf{r}) \in \mathcal{F}\big).
\end{aligned} \tag{10}$$

We examine each of the terms in the right-hand side of the inequality:

$$\begin{aligned}
&\mathfrak{R}_m\big((x, y) \mapsto \mathbf{r}(x) - y\mathbf{h}(x) \colon (\mathbf{h}, \mathbf{r}) \in \mathcal{F}\big) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i(\mathbf{r}(x_i) - y_i\mathbf{h}(x_i))\Big] \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i\mathbf{r}(x_i)\Big] + \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i(y_i\mathbf{h}(x_i))\Big] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i\mathbf{r}(x_i)\Big] + \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\Big[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i\mathbf{h}(x_i)\Big] \\
&= \mathfrak{R}_m(\mathcal{R}) + \mathfrak{R}_m(\mathcal{H}),
\end{aligned}$$

since $-y_i\sigma_i$ and σ_i are distributed in the same way, we effectively can absorb $-y_i$ into the definition of σ_i. Lastly, since the $\boldsymbol{\alpha}$ does not affect the Rademacher complexity, we have that

$\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i \mathbf{h}(x_i)\right] = \mathfrak{R}_m(\mathcal{H})$ and similarly $\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i \mathbf{r}(x_i)\right] = \mathfrak{R}_m(\mathcal{R})$. By a similar reasoning, we also have that

$$\mathfrak{R}_m\big((x,y)\mapsto -\mathbf{r}(x)\colon (\mathbf{h},\mathbf{r})\in\mathcal{F}\big) = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{(\mathbf{h},\mathbf{r})\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \sigma_i \mathbf{r}(x_i)\right] = \mathfrak{R}_m(\mathcal{R}).$$

Combining the above, we have that the right-hand side of Inequality 10 is bounded as follows

$$\mathfrak{R}_m(\mathcal{L}_{\mathrm{MB},\mathcal{F}}) \le a\ \Phi_1'(\Phi_1^{-1}(1))\mathfrak{R}_m(\mathcal{H}) + (c\,b\ \Phi_2'\big(\Phi_2^{-1}(1/c)\big) + a\ \Phi_1'(\Phi_1^{-1}(1)))\mathfrak{R}_m(\mathcal{R}),$$

which completes the proof. □

By taking $\Phi_1(u) = \Phi_2(u) = \exp(u)$, we have the following theorem since in this case, we simply have that $C_{\Phi_1} = 2a$ and $C_{\Phi_2} = 2b$.

Theorem 6. *Let $\mathcal{H}$ and $\mathcal{R}$ be family of functions mapping $\mathcal{X}$ to $\mathbb{R}$. Assume $N > 1$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of a sample S of size m from $\mathcal{D}$, the following holds for all $(\mathbf{h},\mathbf{r}) \in \mathcal{F}$:*

$$R(\mathbf{h},\mathbf{r}) \le \mathop{\mathbb{E}}_{(x,y)\sim S}[L_{\mathrm{MB}}(\mathbf{h},\mathbf{r},x,y)] + 2a\,\mathfrak{R}_m(\mathcal{H}) + 2(a+b)\mathfrak{R}_m(\mathcal{R}) + \sqrt{\frac{\log 1/\delta}{2m}}.$$

The corollary below is a direct consequence of the above Theorem 6 and it presents margin-based guarantees that are subsequently used to derive the BA algorithm.

Corollary 3. *Assume $N > 1$ and fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of an i.i.d. sample S of size m from $\mathcal{D}$, the following holds for all $(\mathbf{h},\mathbf{r}) \in \mathcal{F}$:*

$$R(\mathbf{h},\mathbf{r}) \le \mathop{\mathbb{E}}_{(x,y)\sim S}[L^{\rho}_{\mathrm{MB}}(\mathbf{h},\mathbf{r},x,y)] + \frac{2a}{\rho}\mathfrak{R}_m(\mathcal{H}) + \frac{2(a+b)}{\rho}\mathfrak{R}_m(\mathcal{R}) + \sqrt{\frac{\log 1/\delta}{2m}}.$$

D Direction and step of projected coordinate descent

In this appendix, we provide the details of the projected coordinate descent, projected CD, algorithm by first deriving the direction and then the optimal step. We give a closed form solution of the step size for exponential loss $\Phi(u) = \exp(u)$ and logistic loss $\Phi(u) = \log_2(1+e^u)$.

D.1 Direction

At each iteration $t-1$, the direction $\mathbf{e}_k$ selected by projected CD is $k = \operatorname{argmax}_{j\in[1,N]} |F'(\boldsymbol{\alpha}_{t-1},\mathbf{e}_j)|$ where the derivative is given by the following

$$\begin{aligned} F'(\boldsymbol{\alpha}_{t-1},\mathbf{e}_j) =& \frac{1}{m}\sum_{i=1}^m \Big([r_j(x_i) - y_i h_j(x_i)]\Phi'\big(\mathbf{r}_{t-1}(x_i) - y_i\mathbf{h}_{t-1}(x_i)\big) \\ & - cb\, r_j(x_i)\Phi'\big(-b\,\mathbf{r}_{t-1}(x_i)\big)\Big) + \beta. \end{aligned}$$

Using the definition of $D(i,1)$ and $D(i,2)$, we re-write the derivative as follows:

$$\begin{aligned} F'(\boldsymbol{\alpha}_{t-1},\mathbf{e}_j) =& \frac{Z_t}{m}\sum_{i=1}^m \Big([r_j(x_i) - y_i h_j(x_i)]D_t(i,1) - cb\, r_j(x_i) D_t(i,2)\Big) + \beta \\ =& \frac{Z_t}{m}\Big(2Z_{1,t}\epsilon_{s,j} - Z_{1,t} + Z_{1,t}\overline{r}_{j,1} - cb\, Z_{2,t}\overline{r}_{j,2}\Big) + \beta. \end{aligned}$$

Hence, we have that the descent direction is $k = \operatorname{argmin}_{j\in[1,N]} 2Z_{1,t}\epsilon_{t,j} + Z_{1,t}\overline{r}_{j,1} - cb\, Z_{2,t}\overline{r}_{j,2}$.

D.2 Step

The optimal step values η for direction $\mathbf{e}_k$ is given by $\operatorname{argmin}_{\eta+\alpha_{t-1,k}\geq 0} F(\boldsymbol{\alpha}_{t-1}+\eta\mathbf{e}_k)$. The values η may be found via line search or other numerical methods, but below we derive a closed-form solution by minimizing an upper bound of $F(\boldsymbol{\alpha}_{t-1}+\eta\mathbf{e}_k)$.

Since Φ is convex and since for all $i\in[1,m]$

$$-y_i h_k(x_i)+r_k(x_i)=\frac{1+y_i h_k(x_i)-r_k(x_i)}{2}\cdot(-1)+\frac{1-y_i h_k(x_i)+r_k(x_i)}{2}\cdot(1),$$

we have that the following holds for all $\eta\in\mathbb{R}$

$$\begin{aligned}&\Phi\big(\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)-\eta y_i h_k(x_i)+\eta r_k(x_i)\big)\\ &\leq\frac{1+y_i h_k(x_i)-r_k(x_i)}{2}\Phi\big(\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)-\eta\big)\\ &+\frac{1-y_i h_k(x_i)+r_k(x_i)}{2}\Phi\big(\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)+\eta\big).\end{aligned}$$

Similarly, we have that $-b\,r_k(x_i)=\frac{-b\,r_k(x_i)}{2}\cdot(1)+\frac{b\,r_k(x_i)}{2}\cdot(-1)$

$$\begin{aligned}&\Phi\big(-b\,\mathbf{r}_{t-1}(x_i)-b\,\eta r_k(x_i)\big)\\ &\leq\frac{-b\,r_k(x_i)}{2}\Phi\big(-b\,\mathbf{r}_{t-1}(x_i)+\eta\big)+\frac{b\,r_k(x_i)}{2}\Phi\big(-b\,\mathbf{r}_{t-1}(x_i)-\eta\big)\end{aligned}$$

Thus, we can upper-bound F as follows:

$$\begin{aligned}F(\boldsymbol{\alpha}_{t-1}+\eta\mathbf{e}_k)\leq&\frac{1}{m}\sum_{i=1}^{m}\frac{1+y_i h_k(x_i)-r_k(x_i)}{2}\Phi\big(\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)-\eta\big)\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{1-y_i h_k(x_i)+r_k(x_i)}{2}\Phi\big(\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)+\eta\big)\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{-b\,r_k(x_i)}{2}c\,\Phi\big(-b\,\mathbf{r}_{t-1}(x_i)+\eta\big)\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{b\,r_k(x_i)}{2}c\,\Phi\big(-b\,\mathbf{r}_{t-1}(x_i)-\eta\big)+\sum_{j=1}^{N}\alpha_{t-1}\beta+\beta\eta\end{aligned}$$

We define $J(\eta)$ to be the right-hand side of the inequality above. We will select η as the solution of $\min_{\eta+\alpha_{t-1,k}\geq 0} J(\eta)$, which is a convex optimization problem since J is convex.

D.2.1 Exponential loss

When $\Phi(u)=\exp(u)$, the J function is given by

$$\begin{aligned}J(\eta)=&\frac{1}{m}\sum_{i=1}^{m}\frac{1+y_i h_k(x_i)-r_k(x_i)}{2}e^{\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)}e^{-\eta}\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{1-y_i h_k(x_i)+r_k(x_i)}{2}e^{\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)}e^{\eta}\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{-b\,r_k(x_i)}{2}ce^{-b\,\mathbf{r}_{t-1}(x_i)}e^{\eta}\\ &+\frac{1}{m}\sum_{i=1}^{m}\frac{b\,r_k(x_i)}{2}ce^{-b\,\mathbf{r}_{t-1}(x_i)}e^{-\eta}+\sum_{j=1}^{N}\alpha_{t-1}\beta+\beta\eta.\end{aligned}$$

Since $e^{\mathbf{r}_{t-1}(x_i)-y_i\mathbf{h}_{t-1}(x_i)} = \Phi'\big(\mathbf{r}_{t-1}(x_i) - y_i\mathbf{h}_{t-1}(x_i)\big) = Z_t D_t(i,1)$ and $e^{-b\,\mathbf{r}_{t-1}(x_i)} = \Phi'\big(-b\,\mathbf{r}_{t-1}(x_i)\big) = Z_t D_t(i,2)$, it implies that

$$J(\eta) = \frac{Z_t}{m}\Big((1-\epsilon_{t,k}-\frac{\overline{r}_{k,1}}{2})Z_{1,t}e^{-\eta} + (\epsilon_{t,k}+\frac{\overline{r}_{k,1}}{2})Z_{1,t}e^{\eta} + \frac{-b\,\overline{r}_{k,2}}{2}cZ_{2,t}e^{\eta} + \frac{b\,\overline{r}_{k,2}}{2}cZ_{2,t}e^{-\eta}\Big) + \sum_{j=1}^{N}\alpha_{t-1}\beta + \beta\eta.$$

For simplicity below, we define $A = Z_{1,t}(1-\epsilon_{t,k}-\frac{\overline{r}_{k,1}}{2}) + cZ_{2,t}\frac{b\,\overline{r}_{k,2}}{2}$ and $Z = Z_{1,t}(\epsilon_{t,k}+\frac{\overline{r}_{k,1}}{2}) + cZ_{2,t}\frac{-b\,\overline{r}_{k,2}}{2}$ so that J can be written as

$$J(\eta) = \frac{Z_t}{m}\Big(Ae^{-\eta} + Ze^{\eta}\Big) + \sum_{j=1}^{N}\alpha_{t-1}\beta + \beta\eta.$$

Introducing a Lagrange variable $\lambda \geq 0$, the optimization problem then becomes

$$L(\eta,\lambda) = J(\eta) - \lambda(\eta + \alpha_{t-1,k}) \text{ with } \nabla_\eta L(\eta,\lambda) = J'(\eta) - \lambda.$$

By the KKT conditions, at the solution (η^*,λ^*), $J'(\eta^*) = \lambda^*$ and $\lambda^*(\eta^* + \alpha_{t-1,k}) = 0$. Thus, we can fall in one of the two following cases:

1. $(\lambda^* > 0) \Leftrightarrow (J'(\eta^*) > 0)$ and $\eta^* = -\alpha_{t-1,k}$
2. $\lambda^* = 0$ and η^* is a solution of the equation $J(\eta^*) = 0$

The first case can be written as

$$\frac{Z_t}{m}\Big(-Ae^{\alpha_{t-1,k}} + Ze^{-\alpha_{t-1,k}}\Big) + \beta > 0 \Leftrightarrow Ae^{\alpha_{t-1,k}} - Ze^{-\alpha_{t-1,k}} < \frac{m}{Z_t}\beta.$$

For the second case we have to solve $J'(\eta) = 0$ which can be written as $e^{2\eta} + \frac{m\beta}{Z_t Z}e^{\eta} - \frac{A}{Z}$. The solution is given by

$$e^{\eta} = -\frac{m\beta}{2Z_t Z} + \sqrt{\Big(\frac{m\beta}{2Z_t Z}\Big)^2 + \frac{A}{Z}} \Leftrightarrow \eta = \log\Big[-\frac{m\beta}{2Z_t Z} + \sqrt{\Big(\frac{m\beta}{2Z_t Z}\Big)^2 + \frac{A}{Z}}\Big].$$

Noting that $A = Z_{1,t} - Z$, the above can be simplified to

$$\eta = \log\Big[-\frac{m\beta}{2Z_t Z} + \sqrt{\Big(\frac{m\beta}{2Z_t Z}\Big)^2 + \frac{Z_{1,t}}{Z} - 1}\Big]. \tag{11}$$

D.2.2 Logistic loss

For the logistic loss, we have that for any $u \in \mathbb{R}$, $\Phi(-u) = \log_2(1+e^{-u})$ and $\Phi'(-u) = \frac{1}{\log 2(1+e^{u})}$. We have the following upper bound

$$\begin{aligned}\Phi(-u-v) - \Phi(-u) &= \log_2\Big(\frac{1+e^{-u}+e^{-u-v}-e^{-u}}{1+e^{-u}}\Big) = \log_2\Big(1 + \frac{e^{-v}-1}{e^{-u}+1}\Big)\\ &\leq \frac{e^{-v}-1}{\log 2(1+e^{u})} = \Phi'(-u)(e^{-v}-1),\end{aligned}$$

which allows us to write

$$\begin{aligned}F(\boldsymbol{\alpha}_{t-1} + \eta\mathbf{e}_k) - F(\boldsymbol{\alpha}_{t-1}) \leq & \frac{1}{m}\sum_{i=1}^{m}\Phi'(\mathbf{r}_{t-1}(x_i) - y_i\mathbf{h}_{t-1}(x_i))(e^{-\eta y_i h_k(x_i)+\eta r_k(x_i)} - 1)\\ &+ c\,\Phi'(-b\,\mathbf{r}_{t-1}(x_i))(e^{-b\,\eta r_k(x_i)} - 1) + \beta\eta.\end{aligned}$$

From here, we can use a very similar reasoning as the exponential loss which results in a similar expression for the step size.

E Convergence analysis of algorithm

In the section, we prove the convergence of the projected CD algorithm for $F(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^{m} \Phi\big(\mathbf{r}_t(x_i) - y_i \mathbf{h}_t(x_i)\big) + c\,\Phi\big(-b\,\mathbf{r}_t(x_i)\big) + \beta \sum_{j=1}^{N} \alpha_j$.

Theorem 4. *Assume that Φ is twice differentiable and that $\Phi''(u) > 0$ for all $u \in \mathbb{R}$. Then, the projected CD algorithm applied to F converges to the solution $\boldsymbol{\alpha}^*$ of the optimization problem $\max_{\boldsymbol{\alpha} \geq 0} F(\boldsymbol{\alpha})$. If additionally Φ is strongly convex over the path of the iterates $\boldsymbol{\alpha}_t$ then there exists $\tau > 0$ and $\nu > 0$ such that for all $t > \tau$,*

$$F(\boldsymbol{\alpha}_{t+1}) - F(\boldsymbol{\alpha}^*) \leq (1 - \frac{1}{\nu})(F(\boldsymbol{\alpha}_t) - F(\boldsymbol{\alpha}^*)). \tag{12}$$

Proof. Let $\mathbf{H}$ be the matrix in $\mathbb{R}^{2m \times N}$ defined by $\mathbf{H}_{(i,1),j} = y_i h_j(x_i) - r_j(x_i)$ and $\mathbf{H}_{(i,2),j} = b\,r_j(x_i)$ for all $i \in [1, m]$ and for all $j \in [1, N]$, and let $\mathbf{e}_{(i,1)}$ and $\mathbf{e}_{(i,2)}$ be unit vectors in $\mathbb{R}^{2m}$. Then for any $\boldsymbol{\alpha}$, we have that $\mathbf{e}_{i,1}^T \mathbf{H}\boldsymbol{\alpha} = \sum_{j=1}^{N} \alpha_j (y_i h_j(x_i) - r_j(x_i))$ and $\mathbf{e}_{i,2}^T \mathbf{H}\boldsymbol{\alpha} = b \sum_{j=1}^{N} \alpha_j r_j(x_i)$. Thus, we can write for any $\boldsymbol{\alpha} \in \mathbb{R}^N$,

$$F(\boldsymbol{\alpha}) = G(\mathbf{H}\boldsymbol{\alpha}) + \Lambda^T \boldsymbol{\alpha}, \tag{13}$$

where $\Lambda = (\Lambda_1, \ldots, \Lambda_N)^T$ and where G is the function defined by

$$G(\mathbf{u}) = \frac{1}{m}\sum_{i=1}^{m} \Phi(-e_{i,1}^T \mathbf{u}) + c\Phi(-e_{i,2}^T \mathbf{u}) = \frac{1}{m}\sum_{i=1}^{m} \Phi(-\mathbf{u}_{i,1}) + c\Phi(-\mathbf{u}_{i,2}) \tag{14}$$

for all $\mathbf{u} \in \mathbb{R}^{2m}$ with $u_{i,1}$ its $(i,1)$th coordinate and $u_{i,2}$ its $(i,2)$th coordinate. Since Φ is differentiable, the function G is differentiable and $\nabla^2 G(\mathbf{u})$ is a diagonal matrix with diagonal entries $\frac{1}{m}\Phi''(-u_{i,1}) > 0$ or $\frac{c}{m}\Phi''(-u_{i,2}) > 0$ for all $i \in [1, m]$. Thus, $\nabla^2 G(\mathbf{H}\boldsymbol{\alpha})$ is positive definite for all $\boldsymbol{\alpha}$. The conditions of Theorem 2.1 of [20] are therefore satisfied for the optimization problem

$$\min_{\boldsymbol{\alpha} \geq 0} G(\mathbf{H}\boldsymbol{\alpha}) + \Lambda^T \boldsymbol{\alpha}, \tag{15}$$

thereby guaranteeing the convergence of the projected CD method applied to F. If additionally F is strongly convex over the sequence of $\boldsymbol{\alpha}_t$s, the by the result of [20][page 26], the Inequality 12 holds for the projected coordinate method that we are using which selects the best direction at each round, as with the Gauss-Southwell method. □

F Calibration

In this section, we show that $L_{\text{SB}}(h, r, x, y) = e^{a\,(r(x) - yh(x))} + ce^{-b\,r(x)}$ is a calibrated loss whenever $\frac{b}{a} = 2\sqrt{\frac{1-c}{c}}$. Below, let $L := L_{\text{SB}}(h, r, x, y)$ and define $\eta(x) = \mathbb{P}(Y = +1 | X = x)$.

Theorem 1. *For $a > 0$ and $b > 0$, the $\inf_{(h,r)} \mathbb{E}_{(x,y)}[L(h, r, x, y)]$ is attained at (h_L^*, r_L^*) such that $\text{sign}(h^*) = \text{sign}(h_L^*)$ and $\text{sign}(r^*) = \text{sign}(r_L^*)$ if and only if $\frac{b}{a} = 2\sqrt{\frac{1-c}{c}}$.*

Proof. Conditioning on the label y, we can write the generalization error for the $L(h, r, x, y)$ as follows

$$\mathop{\mathbb{E}}_{(x,y)}[L(h, r, x, y)] = \mathop{\mathbb{E}}_{x}[\eta(x)\Psi(-h(x), r(x)) + (1 - \eta(x))\Psi(h(x), r(x))],$$

where $\Psi(-h(x), r(x)) = e^{a\,(r(x) - h(x))} + ce^{-b\,r(x)}$. For simplicity, we also let $L_\Psi(h(x), r(x)) = \eta(x)\Psi(-h(x), r(x)) + (1 - \eta(x))\Psi(h(x), r(x))$. Since the infimum is over all measurable functions $(h(x), r(x))$, we have that $\inf_{(h,r)} \mathbb{E}_x L_\Psi(h(x), r(x)) = \mathbb{E}_x \inf_{(h(x), r(x))} L_\Psi(h(x), r(x))$. Thus, we need to find the optimal (u, v) for a fixed x that minimizes $L_\Psi(u, v)$ over all measurable functions, which is a convex optimization problem. When $\eta(x) = 0$, the sign of the minimizers of $L_\Psi(u, v)$ are $u^* < 0$ and $v^* > 0$ while when $\eta(x) = 1$, the the sign of the minimizers are $u^* > 0$ and $v^* > 0$, which matches the sign of h^* and r^* in both cases respectively. Now for $\eta(x) \in]0, 1[$, we take the derivative of $L_\Psi(u, v)$ with respect to u

$$\frac{\partial L_\Psi(u,v)}{\partial u} = -\eta(x) a\, e^{a\,(v-u)} + (1 - \eta(x)) a\, e^{a\,(u+v)}.$$

Setting it to zero and solving for u, we have that $u^* = \frac{1}{2a}\log(\frac{\eta(x)}{1-\eta(x)})$. We can now see that $u^* > 0$ if $\eta(x) > \frac{1}{2}$ and $u^* \leq 0$ if $\eta(x) \leq \frac{1}{2}$. Recalling that $h^* = \eta(x) - \frac{1}{2}$, we can conclude that the sign of u^* matches the sign of h^*.

We now take the derivative of $L_\Psi(u^*, v)$ with respect to v

$$\frac{\partial L_\Psi(u^*,v)}{\partial v} = \eta(x)e^{a\,(v-u^*)} + (1-\eta(x))e^{a\,(v+u^*)} + c(-b\,)e^{-b\,v}.$$

Setting it equal to zero and using the fact that $\eta(x)e^{-a\,u^*} + (1-\eta(x))e^{a\,u^*} = 2\sqrt{\eta(x)(1-\eta(x))}$, we have that

$$v^* = \frac{1}{a+b}\log\left(\frac{cb}{2a}\sqrt{\frac{1}{\eta(x)(1-\eta(x))}}\right).$$

Now, we know that the Bayes classifiers (h^*, r^*) satisfy $h^* = \eta(x) - \frac{1}{2}$ and $r^* = |h^*| - \frac{1}{2} + c$ so that the following holds

$$\eta(x)(1-\eta(x)) = \frac{1}{4} - (h^*)^2 = \frac{1}{4} - (r^* + \frac{1}{2} - c)^2.$$

Thus, we can replace $\eta(x)(1-\eta(x))$ in the definition of v^* to arrive at this equation

$$v^* = \frac{1}{a+b}\log\left(\frac{cb}{2a}\sqrt{\frac{1}{\frac{1}{4}-(r^*+\frac{1}{2}-c)^2}}\right).$$

We now analyze when $v^* > 0$ which is equivalent to

$$\frac{1}{a+b}\log\left(\frac{cb}{2a}\sqrt{\frac{1}{\frac{1}{4}-(r^*+\frac{1}{2}-c)^2}}\right) > 0 \Leftrightarrow \frac{cb}{2a}\sqrt{\frac{1}{\frac{1}{4}-(r^*+\frac{1}{2}-c)^2}} > 1$$
$$\Leftrightarrow \frac{cb}{2a} > \sqrt{\frac{1}{4} - (r^* + \frac{1}{2} - c)^2}.$$

Since $\sqrt{\frac{1}{4} - (\frac{1}{2} - c)^2} > \sqrt{\frac{1}{4} - (r^* + \frac{1}{2} - c)^2}$ for $r^* > 0$ and using the fact that $c(1-c) = \frac{1}{4} - (\frac{1}{2} - c)^2$, we need that $\frac{cb}{2a} \geq \sqrt{c(1-c)}$. By similar reasoning for $v^* \leq 0$, we need that $\frac{cb}{2a} \leq \sqrt{c(1-c)}$. Thus, we can conclude that the sign of v^* matches the sign of r^* if and only if $\frac{cb}{2a} = \sqrt{c(1-c)}$. □

G Abstention stumps

Under the assumptions of Section 4.3, the derivative of F can be simplified as follows

$$F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j) = \frac{Z_t}{m}\Big(-\sum_{i:y_i h_j(x_i)=+1} D_t(i,1) + \sum_{i:y_i h_j(x_i)=-1} D_t(i,1) + \sum_{i:r_j(x_i)=1} D_t(i,1)$$
$$- cb\sum_{i:r_j(x_i)=1} D_t(i,2)\Big) + \beta \tag{16}$$

From the definition of $D(i,1)$ and the assumptions on $h(x)$ and $r(x)$, the following holds

$$\sum_{i=1}^{m} D_t(i,1) = \sum_{i:y_i h_j(x_i)=+1} D_t(i,1) + \sum_{i:y_i h_j(x_i)=-1} D_t(i,1) + \sum_{i:r_j(x_i)=1} D_t(i,1)$$

Solving for $\sum_{i:y_i h_j(x_i)=+1} D_t(i,1)$ and plugging it in Equation 16, we can simplify the derivative

$$F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j) = \frac{Z_t}{m}\Big(2\sum_{i:y_i h_j(x_i)=-1} D_t(i,1) + 2\sum_{i:r_j(x_i)=1} D_t(i,1) - \sum_{i=1}^{m} D_t(i,1) - cb\sum_{i:r_j(x_i)=1} D_t(i,2)\Big) + \beta$$
$$= \frac{Z_t}{m}\Big(2Z_{1,t}\epsilon_{t,j} + 2Z_{1,t}\overline{r}_{j,1} - cb\,Z_{2,t}\overline{r}_{j,2} - Z_{1,t}\Big) + \beta$$

Thus, the optimal descent direction is $k = \operatorname{argmin}_{j\in[1,N]} 2Z_{1,t}\epsilon_{t,j} + 2Z_{1,t}\overline{r}_{j,1} - cb\,Z_{2,t}\overline{r}_{j,2}$

Below, we provide the proof of the lemma that was needed to decouple the optimization problem for the abstention stumps.

Lemma 5. *The optimization problem without the constraint* $(\theta_1 < \theta_2)$ *can be decomposed as follows:*

$$\begin{aligned}
&\operatorname*{argmin}_{\theta_1,\theta_2}\sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i\le\theta_1} + 1_{y_i=-1}1_{X_i>\theta_2}] + \big(2D_t(i,1) - cb\,D_t(i,2)\big)1_{\theta_1<X_i\le\theta_2}.\\
&= \operatorname*{argmin}_{\theta_1}\sum_{i=1}^{m} 2D_t(i,1)1_{y_i=+1}1_{X_i\le\theta_1} + \big(2D_t(i,1) - cb\,D_t(i,2)\big)1_{\theta_1<X_i}\\
&+ \operatorname*{argmin}_{\theta_2}\sum_{i=1}^{m} 2D_t(i,1)1_{y_i=-1}1_{X_i>\theta_2} + \big(2D_t(i,1) - cb\,D_t(i,2)\big)1_{X_i\le\theta_2}.
\end{aligned}$$

Proof. For simplicity below, let $\kappa = \big(2D_t(i,1) - cb\,D_t(i,2)\big)$ and observe that the following identity holds:

$$1_{\theta_1<X_i\le\theta_2} = 1_{\theta_1<X_i} + 1_{X_i\le\theta_2} - 1$$

In view of that, we can write

$$\begin{aligned}
&\operatorname*{argmin}_{\theta_1,\theta_2}\sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i\le\theta_1} + 1_{y_i=-1}1_{X_i>\theta_2}] + \kappa 1_{\theta_1<X_i\le\theta_2}\\
&= \operatorname*{argmin}_{\theta_1,\theta_2}\sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i\le\theta_1} + 1_{y_i=-1}1_{X_i>\theta_2}] + \kappa[1_{\theta_1<X_i} + 1_{X_i\le\theta_2} - 1]\\
&= \operatorname*{argmin}_{\theta_1,\theta_2}\sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i\le\theta_1} + 1_{y_i=-1}1_{X_i>\theta_2}] + \kappa 1_{\theta_1<X_i}\\
&\qquad\qquad + \kappa 1_{X_i\le\theta_2} - \kappa\\
&= \operatorname*{argmin}_{\theta_1,\theta_2}\sum_{i=1}^{m} 2D_t(i,1)[1_{y_i=+1}1_{X_i\le\theta_1} + 1_{y_i=-1}1_{X_i>\theta_2}] + \kappa 1_{\theta_1<X_i}\\
&\qquad\qquad + \kappa 1_{X_i\le\theta_2}\\
&= \operatorname*{argmin}_{\theta_1}\sum_{i=1}^{m} 2D_t(i,1)1_{y_i=+1}1_{X_i\le\theta_1} + \kappa 1_{\theta_1<X_i}\\
&+ \operatorname*{argmin}_{\theta_2}\sum_{i=1}^{m} 2D_t(i,1)1_{y_i=-1}1_{X_i>\theta_2} + \kappa 1_{X_i\le\theta_2}.
\end{aligned}$$

□

H Alternative surrogate, L_{MB}

In this section, we derive the boosting algorithm for the surrogate loss

$$L_{\text{MB}}(\mathbf{h},\mathbf{r},x,y) = \max\Big(\Phi_1\big(a\,[\mathbf{r}(x) - y\mathbf{h}(x)]\big), c\,\Phi_2\big(-b\,\mathbf{r}(x)\big)\Big). \tag{17}$$

By a similar reasoning as Section 4, the objective function $F(\boldsymbol{\alpha})$ of our optimization problem is given by the following

$$F(\boldsymbol{\alpha}) = \frac{1}{m}\sum_{i=1}^{m}\max\Big(e^{a\,[\mathbf{r}_t(x_i) - y_i\mathbf{h}_t(x_i)]}, cb\,e^{-b\,\mathbf{r}_t(x_i)}\Big) + \beta\sum_{j=1}^{N}\alpha_j.$$

For simplicity, we define $u_t(i) = e^{a\,[\mathbf{r}_t(x_i) - y_i\mathbf{h}_t(x_i)]}$, $v_t(i) = cb\,e^{-b\,\mathbf{r}_t(x_i)}$ and $w_t(i) = \max(u_t(i), v_t(i))$. We also let $1_{u_t(i)}$ be the indicator functions that equals 1 if $u_t(i) \ge v_t(i)$ and similarly $1_{v_t(i)}$ be the indicator functions that equals 1 if $v_t(i) > u_t(i)$. For any $t \in [1, T]$, we also define the distribution

$$\mathcal{D}_t(i) = \frac{w_{t-1}(i)}{Z_t}, \tag{18}$$

where Z_t is the normalization factor given by $Z_t = \sum_{i=1}^m w_{t-1}(i)$.

We then apply projected coordinate descent to this objective function. Notice that our objective F is differentiable everywhere except when $u_t(i) = v_t(i)$. A true maximum descent algorithm would choose the element of the subgradient that is closest to 0 as the descent direction. However, since this event is rare in our case, we arbitrarily a pick descent direction that is an element of the subgradient. For simplicity below, we will use the symbol $F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j)$ to denote the directional derivative with the added condition that for the non-differentiable point, we choose the direction that is an element of the subgradient.

H.0.3 Direction and step

At each iteration $t-1$, the direction $\mathbf{e}_k$ selected by projected CD is $k = \operatorname{argmax}_{j\in[1,N]} |F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j)|$ where

$$\begin{aligned}
&F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j) \\
&= \frac{1}{m}\sum_{i=1}^m \Big(a\,[-y_i h_j(x_i) + r_j(x_i)]1_{u_{t-1}(i)} - cb\, r_j(x_i)1_{v_{t-1}(i)}\Big) w_{t-1}(i) + \beta \\
&= \frac{1}{m}\sum_{i=1}^m \Big(-a\, y_i h_j(x_i)1_{u_{t-1}(i)} - [-(a\ + cb\,)1_{u_{t-1}(i)} + cb\,] r_j(x_i)\Big) w_{t-1}(i) + \beta \\
&= \frac{1}{m}\sum_{i=1}^m \Big(-a\, y_i h_j(x_i)1_{u_{t-1}(i)} - [-(a\ + cb\,)1_{u_{t-1}(i)} + cb\,] r_j(x_i)\Big) \mathcal{D}_t(i) Z_t + \beta. \qquad (19)
\end{aligned}$$

The step can simply be found via line search or other numerical methods.

H.1 Abstention stumps

We focus in on a special case where the base classifiers have a specific form defined as follows: $h(x)$ takes values in $\{-1, 0, 1\}$ and $r(x)$ take values in $\{0, 1\}$. We also have the added the condition that for each sample point x, only one of the two components of $(h(x), r(x))$ is non-zero. Under this setting, Equation 19 can be simplified as follows. The derivative of F is given by

$$F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j) = \frac{1}{m}\sum_{i=1}^m \Big(a\,[-y_i h_j(x_i) + r_j(x_i)]1_{u_{t-1}(i)} - cb\, r_j(x_i)1_{v_{t-1}(i)}\Big) D_t(i) Z_t + \beta,$$

which can be rewritten as

$$\begin{aligned}
&= \frac{Z_t}{m}\Big(a\,[\sum_{i:y_i h_j(x_i)=-1} 1_{u_{t-1}(i)} D_t(i) - \sum_{i:y_i h_j(x_i)=+1} 1_{u_{t-1}(i)} D_t(i) + \sum_{i:r_j(x_i)=1} 1_{u_{t-1}(i)} D_t(i)] \\
&- cb \sum_{i:r_j(x_i)=1} D_t(i) 1_{v_{t-1}(i)}\Big) + \beta. \qquad (20)
\end{aligned}$$

From the assumptions on $h(x)$ and $r(x)$, the relation below holds:

$$\sum_{i=1}^m D_t(i)1_{u_{t-1}(i)} = \sum_{i:y_i h_j(x_i)=+1} D_t(i)1_{u_{t-1}(i)} + \sum_{i:y_i h_j(x_i)=-1} D_t(i)1_{u_{t-1}(i)} + \sum_{i:r_j(x_i)=1} D_t(i)1_{u_{t-1}(i)},$$

which is equivalent to the following

$$-\sum_{i:y_i h_j(x_i)=+1} D_t(i)1_{u_{t-1}(i)} = \sum_{i:y_i h_j(x_i)=-1} D_t(i)1_{u_{t-1}(i)} + \sum_{i:r_j(x_i)=1} D_t(i)1_{u_{t-1}(i)} - \sum_{i=1}^m D_t(i)1_{u_{t-1}(i)}.$$

Table 1: For each data set, we report the sample size and the number of features.

Data Sets	Sample Size	Feature
`australian`	690	14
`cod`	369	8
`skin`	400	3
`banknote`	1,372	4
`haberman`	306	3
`pima`	768	8

Plugging this into equation 20, we have that

$$\begin{aligned} &F'(\boldsymbol{\alpha}_{t-1}, \mathbf{e}_j) \\ &= \frac{Z_t}{m}\Big(a\,[2\sum_{i:y_i h_j(x_i)=-1} 1_{u_{t-1}(i)} D_t(i) + 2\sum_{i:r_j(x_i)=1} 1_{u_{t-1}(i)} D_t(i) - \sum_{i=1}^{m} 1_{u_{t-1}(i)} D_t(i)] \\ &- cb\sum_{i:r_j(x_i)=1} D_t(i) 1_{v_{t-1}(i)}\Big) + \beta. \end{aligned} \tag{21}$$

This in turn implies that our weak learning algorithm is given by the following:

$$l(\theta_1, \theta_2) = \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)}[1_{y_i=+1} 1_{h_{\theta_1,\theta_2}(x)=-1} + 1_{y_i=-1} 1_{h_{\theta_1,\theta_2}(x)=1}] + [2a\, 1_{u(i)} - cb\, 1_{v(i)}]\; 1_{r_{\theta_1,\theta_2}(x)=1}].$$

The following lemma allows us to decouple the optimization problem into two optimization problems with respect to θ_1 and θ_2 that can be solved in linear time.

Lemma 7. *The optimization problem without the constraint* $(\theta_1 < \theta_2)$ *can be decomposed as follows:*

$$\begin{aligned} &\mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}\big(2a\, 1_{u(i)}[1_{y_i=+1} 1_{X_i\le\theta_1} + 1_{y_i=-1} 1_{X_i>\theta_2}] + [2a\, 1_{u(i)} - cb\, 1_{v(i)}] 1_{\theta_1<X_i\le\theta_2}\big) \\ &= \mathop{\mathrm{argmin}}_{\theta_1} \mathop{\mathbb{E}}_{i\sim D}\big(2a\, 1_{u(i)} 1_{y_i=+1} 1_{X_i\le\theta_1} + [2a\, 1_{u(i)} - cb\, 1_{v(i)}] 1_{\theta_1<X_i}\big) \\ &+ \mathop{\mathrm{argmin}}_{\theta_2} \mathop{\mathbb{E}}_{i\sim D}\big(2a\, 1_{u(i)} 1_{y_i=-1} 1_{X_i>\theta_2} + [2a\, 1_{u(i)} - cb\, 1_{v(i)}] 1_{X_i\le\theta_2}\big). \end{aligned}$$

Proof. For simplicity, let $\kappa = 2a\, 1_{u(i)} - cb\, 1_{v(i)}$ and observe that the following identity holds:

$$1_{\theta_1<X_i\le\theta_2} = 1_{\theta_1<X_i} + 1_{X_i\le\theta_2} - 1$$

In view of that, we can write

$$\begin{aligned} &\mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)}[1_{y_i=+1} 1_{X_i\le\theta_1} + 1_{y_i=-1} 1_{X_i>\theta_2}] + \kappa 1_{\theta_1<X_i\le\theta_2}] \\ &= \mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)}[1_{y_i=+1} 1_{X_i\le\theta_1} + 1_{y_i=-1} 1_{X_i>\theta_2}] + \kappa(1_{\theta_1<X_i} + 1_{X_i\le\theta_2} - 1)] \\ &= \mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)} 1_{y_i=+1} 1_{X_i\le\theta_1} + \kappa 1_{\theta_1<X_i} + 2a\, 1_{u(i)} 1_{y_i=-1} 1_{X_i>\theta_2} \\ &\qquad\qquad + \kappa 1_{X_i\le\theta_2} - \kappa] \\ &= \mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)} 1_{y_i=+1} 1_{X_i\le\theta_1} + \kappa 1_{\theta_1<X_i}] \\ &\quad + \mathop{\mathrm{argmin}}_{\theta_1,\theta_2} \mathop{\mathbb{E}}_{i\sim D}[2a\, 1_{u(i)} 1_{y_i=-1} 1_{X_i>\theta_2} + \kappa 1_{X_i\le\theta_2}] \end{aligned}$$

which completes the proof. □

I Data sets

Table 1 shows the sample size and number of features for each data set used in our experiments.

J Experiments

In this appendix, we report the results of several experiments by presenting different tables in order to compare the three algorithms studied in this paper: TSB, DHL, and BA. In each table, we provide the average and standard deviation on the test set for the hyper parameter configurations that admitted the smallest abstention loss on the validation set. Overall, these results reveal that BA yields a significant improvement in practice for all the data sets across different values of cost c.

Table 2 gives the average abstention loss on the test set for TSB, DHL, and BA algorithms. Across almost all the different values of cost c, the BA algorithm attains the smallest abstention loss compared with the TSB and DHL algorithms. On some datasets, the TSB performs better than the DHL algorithm, but on other datasets, its performance largely deteriorates. We also see that the effects of changing the cost c of rejection for some datasets is much stronger than for other datasets. For example, the `pima` dataset has a large change in abstention loss as c increases while for `banknote` dataset the difference in abstention loss is very small. These changes reflect the changes in the fraction of points rejected by the algorithms, see Table 3. Note that this effect also depends on the algorithm as seen in the `cod` dataset where BA algorithm's abstention loss changes only slightly while for the other two algorithms the difference is much higher as c increases.

Table 3 shows the fraction of points that are rejected on the test set. For all three algorithms, the fraction of points rejected decreases as the cost c of rejection increases. Moreover, the fraction of points rejected is much higher for some datasets. For most values of c, the DHL algorithm appears to reject less frequently, but its abstention loss is also higher. For `haberman`, `australian`, and `pima` datasets, the TSB algorithm rejection rates is quite high, which reinforces our claim that DHL and BA algorithms are better algorithms. Finally, Table 4, presents the classification loss on non-rejected points for different values of c. As c increases, we see that more points are classified incorrectly, which is in accordance with the previous table since it shows that we are also rejecting less points.

Table 2: Average abstention loss along with the standard deviations on the test set for the TSB Algorithm, DHL Algorithm and BA Algorithm

Cost	skin TSB	skin DHL	skin BA	cod TSB	cod DHL	cod BA
0.05	0.0482 ± 0.0156	0.024 ± 0.016	0.0258 ± 0.0157	0.0384 ± 0.00926	0.044 ± 0.034	0.0386 ± 0.0152
0.1	0.059 ± 0.0345	0.061 ± 0.031	0.0373 ± 0.0166	0.0784 ± 0.0176	0.077 ± 0.028	0.0624 ± 0.00367
0.15	0.0822 ± 0.0141	0.091 ± 0.031	0.0595 ± 0.0174	0.0807 ± 0.0138	0.123 ± 0.030	0.0593 ± 0.0279
0.2	0.052 ± 0.0262	0.128 ± 0.036	0.04 ± 0.0185	0.0903 ± 0.0245	0.175 ± 0.031	0.0654 ± 0.0274
0.25	0.0667 ± 0.0304	0.158 ± 0.041	0.0425 ± 0.0174	0.0831 ± 0.00896	0.204 ± 0.026	0.0676 ± 0.0304
0.3	0.037 ± 0.0226	0.177 ± 0.044	0.0403 ± 0.0162	0.117 ± 0.0151	0.230 ± 0.022	0.0659 ± 0.0285
0.35	0.0593 ± 0.0272	0.204 ± 0.056	0.0477 ± 0.0144	0.11 ± 0.0182	0.259 ± 0.029	0.0581 ± 0.0313
0.4	0.0907 ± 0.0125	0.231 ± 0.067	0.0567 ± 0.0181	0.106 ± 0.0271	0.273 ± 0.026	0.0692 ± 0.0372
0.45	0.0693 ± 0.033	0.215 ± 0.066	0.0525 ± 0.0186	0.12 ± 0.0246	0.276 ± 0.025	0.065 ± 0.0364

Cost	haberman TSB	haberman DHL	haberman BA	pima TSB	pima DHL	pima BA
0.05	0.05 ± 0.0	0.050 ± 0.000	0.05 ± 0.0	0.0512 ± 0.00247	0.068 ± 0.039	0.05 ± 0.0
0.1	0.103 ± 0.00581	0.143 ± 0.027	0.1 ± 0.0	0.106 ± 0.00787	0.176 ± 0.009	0.1 ± 0.0
0.15	0.15 ± 0.000968	0.213 ± 0.037	0.173 ± 0.0458	0.146 ± 0.00567	0.218 ± 0.023	0.157 ± 0.0221
0.2	0.204 ± 0.0101	0.233 ± 0.036	0.214 ± 0.0256	0.195 ± 0.00489	0.238 ± 0.021	0.172 ± 0.00859
0.25	0.25 ± 0.0	0.256 ± 0.027	0.234 ± 0.0238	0.235 ± 0.00669	0.241 ± 0.025	0.19 ± 0.0211
0.3	0.303 ± 0.0147	0.264 ± 0.019	0.244 ± 0.0196	0.285 ± 0.00428	0.247 ± 0.026	0.201 ± 0.0114
0.35	0.34 ± 0.0123	0.261 ± 0.024	0.265 ± 0.0325	0.327 ± 0.00874	0.250 ± 0.027	0.22 ± 0.016
0.4	0.383 ± 0.0192	0.262 ± 0.028	0.272 ± 0.033	0.374 ± 0.0079	0.255 ± 0.028	0.234 ± 0.0134
0.45	0.441 ± 0.0235	0.258 ± 0.022	0.275 ± 0.0301	0.422 ± 0.0114	0.260 ± 0.034	0.249 ± 0.0171

Cost	australian TSB	australian DHL	australian BA	banknote TSB	banknote DHL	banknote BA
0.05	0.0499 ± 0.000145	0.112 ± 0.033	0.0564 ± 0.0117	0.000873 ± 0.00139	0.091 ± 0.059	0.00247 ± 0.00195
0.1	0.0867 ± 0.00455	0.120 ± 0.024	0.0777 ± 0.016	0.00284 ± 0.00299	0.082 ± 0.070	0.00705 ± 0.00632
0.15	0.13 ± 0.00615	0.128 ± 0.025	0.093 ± 0.0155	0.00411 ± 0.00108	0.081 ± 0.076	0.0044 ± 0.00411
0.2	0.168 ± 0.00612	0.130 ± 0.036	0.111 ± 0.0215	0.00131 ± 0.00197	0.049 ± 0.020	0.00611 ± 0.00509
0.25	0.209 ± 0.00898	0.134 ± 0.038	0.12 ± 0.0171	0.000727 ± 0.00068	0.061 ± 0.022	0.00636 ± 0.00381
0.3	0.244 ± 0.0126	0.137 ± 0.038	0.137 ± 0.0252	0.00371 ± 0.00239	0.083 ± 0.025	0.00735 ± 0.00397
0.35	0.294 ± 0.0141	0.141 ± 0.039	0.144 ± 0.0263	0.0096 ± 0.00426	0.087 ± 0.052	0.00833 ± 0.00465
0.4	0.335 ± 0.0254	0.148 ± 0.042	0.151 ± 0.0273	0.00422 ± 0.00281	0.119 ± 0.028	0.00785 ± 0.00492
0.45	0.365 ± 0.0223	0.150 ± 0.046	0.145 ± 0.0337	0.00447 ± 0.0038	0.136 ± 0.027	0.00738 ± 0.00399

Table 3: Average fraction of points rejected along with the standard deviations on the test set for the TSB Algorithm, DHL Algorithm and BA Algorithm

Cost	skin TSB	skin DHL	skin BA	cod TSB	cod DHL	cod BA
0.05	0.497 ± 0.207	0.180 ± 0.044	0.317 ± 0.111	0.605 ± 0.0523	0.170 ± 0.045	0.557 ± 0.0972
0.1	0.0567 ± 0.0226	0.158 ± 0.047	0.173 ± 0.0374	0.189 ± 0.074	0.146 ± 0.049	0.543 ± 0.0785
0.15	0.237 ± 0.13	0.125 ± 0.032	0.13 ± 0.0476	0.376 ± 0.0563	0.132 ± 0.039	0.197 ± 0.0488
0.2	0.0267 ± 0.0271	0.100 ± 0.025	0.0667 ± 0.0279	0.168 ± 0.0236	0.065 ± 0.026	0.0568 ± 0.0447
0.25	0.0267 ± 0.0271	0.092 ± 0.033	0.0633 ± 0.0287	0.127 ± 0.0202	0.038 ± 0.018	0.0432 ± 0.0313
0.3	0.0233 ± 0.0309	0.090 ± 0.051	0.0567 ± 0.0226	0.146 ± 0.0335	0.027 ± 0.021	0.0486 ± 0.0303
0.35	0.0267 ± 0.0309	0.075 ± 0.051	0.06 ± 0.0309	0.168 ± 0.0303	0.014 ± 0.010	0.027 ± 0.0242
0.4	0.06 ± 0.0501	0.032 ± 0.011	0.05 ± 0.035	0.151 ± 0.0405	0.000 ± 0.000	0.0378 ± 0.0262
0.45	0.08 ± 0.0323	0.005 ± 0.007	0.05 ± 0.035	0.159 ± 0.0563	0.000 ± 0.000	0.0243 ± 0.0262

Cost	haberman TSB	haberman DHL	haberman BA	pima TSB	pima DHL	pima BA
0.05	1.0 ± 0.0	1.000 ± 0.000	1.0 ± 0.0	0.999 ± 0.0026	0.884 ± 0.258	1.0 ± 0.0
0.1	0.997 ± 0.00645	0.738 ± 0.183	1.0 ± 0.0	0.927 ± 0.0311	0.304 ± 0.072	1.0 ± 0.0
0.15	0.997 ± 0.00645	0.348 ± 0.123	0.852 ± 0.297	0.925 ± 0.0408	0.143 ± 0.031	0.321 ± 0.0364
0.2	0.939 ± 0.0313	0.148 ± 0.053	0.216 ± 0.0546	0.901 ± 0.043	0.078 ± 0.024	0.33 ± 0.0405
0.25	1.0 ± 0.0	0.039 ± 0.015	0.187 ± 0.0582	0.894 ± 0.038	0.055 ± 0.007	0.249 ± 0.0359
0.3	0.935 ± 0.0177	0.016 ± 0.028	0.255 ± 0.131	0.923 ± 0.0258	0.039 ± 0.015	0.262 ± 0.0343
0.35	0.945 ± 0.0718	0.007 ± 0.015	0.0581 ± 0.06	0.93 ± 0.023	0.038 ± 0.024	0.238 ± 0.0416
0.4	0.9 ± 0.0664	0.007 ± 0.009	0.0516 ± 0.064	0.918 ± 0.0292	0.034 ± 0.014	0.23 ± 0.0404
0.45	0.923 ± 0.0373	0.013 ± 0.014	0.0516 ± 0.064	0.919 ± 0.0369	0.026 ± 0.009	0.219 ± 0.0414

Cost	australian TSB	australian DHL	australian BA	banknote TSB	banknote DHL	banknote BA
0.05	0.999 ± 0.0029	0.151 ± 0.037	0.346 ± 0.0416	0.00291 ± 0.00356	0.799 ± 0.119	0.00582 ± 0.00493
0.1	0.809 ± 0.0566	0.068 ± 0.017	0.328 ± 0.0452	0.00655 ± 0.00424	0.075 ± 0.021	0.00509 ± 0.00291
0.15	0.772 ± 0.0745	0.049 ± 0.019	0.262 ± 0.0436	0.0225 ± 0.00842	0.060 ± 0.006	0.00509 ± 0.00291
0.2	0.765 ± 0.0597	0.036 ± 0.013	0.177 ± 0.027	0.00291 ± 0.00356	0.072 ± 0.014	0.00509 ± 0.00291
0.25	0.794 ± 0.0245	0.030 ± 0.006	0.142 ± 0.046	0.00291 ± 0.00272	0.066 ± 0.016	0.00509 ± 0.00291
0.3	0.793 ± 0.0469	0.030 ± 0.008	0.142 ± 0.0254	0.00509 ± 0.00291	0.058 ± 0.017	0.00509 ± 0.00291
0.35	0.816 ± 0.036	0.025 ± 0.011	0.114 ± 0.0312	0.0233 ± 0.0109	0.041 ± 0.025	0.00509 ± 0.00291
0.4	0.801 ± 0.0899	0.010 ± 0.006	0.0145 ± 0.0152	0.00509 ± 0.00436	0.048 ± 0.005	0.00509 ± 0.00291
0.45	0.801 ± 0.0551	0.004 ± 0.006	0.0812 ± 0.0288	0.00509 ± 0.00291	0.052 ± 0.012	0.00509 ± 0.00291

Table 4: Average classification error on non-rejected points along with the standard deviation for the TSB Algorithm, DHL Algorithm and BA Algorithm

Cost	skin TSB	skin DHL	skin BA	cod TSB	cod DHL	cod BA
0.05	0.0233 ± 0.0249	0.015 ± 0.016	0.01 ± 0.0133	0.00811 ± 0.0108	0.035 ± 0.034	0.0108 ± 0.0132
0.1	0.0533 ± 0.0356	0.045 ± 0.033	0.02 ± 0.0163	0.0595 ± 0.0202	0.062 ± 0.030	0.00811 ± 0.00662
0.15	0.0467 ± 0.0306	0.073 ± 0.031	0.04 ± 0.0133	0.0243 ± 0.0132	0.103 ± 0.031	0.0297 ± 0.0216
0.2	0.0467 ± 0.0245	0.108 ± 0.034	0.0267 ± 0.0226	0.0568 ± 0.0248	0.162 ± 0.030	0.0541 ± 0.0256
0.25	0.06 ± 0.0327	0.135 ± 0.037	0.0267 ± 0.0226	0.0514 ± 0.0101	0.195 ± 0.026	0.0568 ± 0.0262
0.3	0.03 ± 0.0194	0.150 ± 0.035	0.0233 ± 0.0226	0.073 ± 0.0108	0.222 ± 0.023	0.0514 ± 0.0232
0.35	0.05 ± 0.0279	0.178 ± 0.045	0.0267 ± 0.0226	0.0514 ± 0.0232	0.254 ± 0.028	0.0486 ± 0.0265
0.4	0.0667 ± 0.0279	0.218 ± 0.063	0.0367 ± 0.0267	0.0459 ± 0.0251	0.273 ± 0.026	0.0541 ± 0.0308
0.45	0.0333 ± 0.0279	0.212 ± 0.068	0.03 ± 0.0245	0.0486 ± 0.0369	0.276 ± 0.025	0.0541 ± 0.032

Cost	haberman TSB	haberman DHL	haberman BA	pima TSB	pima DHL	pima BA
0.05	0.0 ± 0.0	0.000 ± 0.000	0.0 ± 0.0	0.0013 ± 0.0026	0.023 ± 0.052	0.0 ± 0.0
0.1	0.00323 ± 0.00645	0.069 ± 0.042	0.0 ± 0.0	0.013 ± 0.0109	0.145 ± 0.016	0.0 ± 0.0
0.15	0.0 ± 0.0	0.161 ± 0.054	0.0452 ± 0.0903	0.00779 ± 0.00486	0.196 ± 0.026	0.109 ± 0.0267
0.2	0.0161 ± 0.0144	0.203 ± 0.041	0.171 ± 0.0332	0.0143 ± 0.00954	0.222 ± 0.023	0.106 ± 0.0106
0.25	0.0 ± 0.0	0.246 ± 0.026	0.187 ± 0.0347	0.0117 ± 0.00757	0.227 ± 0.025	0.127 ± 0.0286
0.3	0.0226 ± 0.0194	0.259 ± 0.021	0.168 ± 0.0416	0.00779 ± 0.00636	0.235 ± 0.023	0.122 ± 0.0161
0.35	0.00968 ± 0.0129	0.259 ± 0.027	0.245 ± 0.0524	0.0013 ± 0.0026	0.236 ± 0.024	0.136 ± 0.025
0.4	0.0226 ± 0.0219	0.259 ± 0.027	0.252 ± 0.0573	0.00649 ± 0.0101	0.242 ± 0.029	0.142 ± 0.0199
0.45	0.0258 ± 0.0079	0.252 ± 0.025	0.252 ± 0.0573	0.00779 ± 0.00636	0.248 ± 0.036	0.151 ± 0.0226

Cost	australian TSB	australian DHL	australian BA	banknote TSB	banknote DHL	banknote BA
0.05	0.0 ± 0.0	0.104 ± 0.033	0.0391 ± 0.00983	0.000727 ± 0.00145	0.051 ± 0.061	0.00218 ± 0.00178
0.1	0.0058 ± 0.00542	0.113 ± 0.023	0.0449 ± 0.0141	0.00218 ± 0.00291	0.074 ± 0.070	0.00655 ± 0.00626
0.15	0.0145 ± 0.00648	0.120 ± 0.023	0.0536 ± 0.0126	0.000727 ± 0.00145	0.072 ± 0.076	0.00364 ± 0.00398
0.2	0.0145 ± 0.0102	0.123 ± 0.037	0.0754 ± 0.0208	0.000727 ± 0.00145	0.034 ± 0.017	0.00509 ± 0.00493
0.25	0.0101 ± 0.0058	0.126 ± 0.037	0.0841 ± 0.0087	0.0 ± 0.0	0.045 ± 0.018	0.00509 ± 0.00371
0.3	0.0058 ± 0.0029	0.128 ± 0.036	0.0942 ± 0.02	0.00218 ± 0.00291	0.066 ± 0.021	0.00582 ± 0.00371
0.35	0.0087 ± 0.0029	0.132 ± 0.036	0.104 ± 0.0187	0.00145 ± 0.00178	0.072 ± 0.044	0.00655 ± 0.00424
0.4	0.0145 ± 0.0112	0.143 ± 0.041	0.145 ± 0.0271	0.00218 ± 0.00178	0.100 ± 0.028	0.00582 ± 0.00436
0.45	0.00435 ± 0.0058	0.148 ± 0.044	0.109 ± 0.0314	0.00218 ± 0.00291	0.112 ± 0.024	0.00509 ± 0.00371