On Discriminative Learning of Prediction Uncertainty

Vojtech Franc [1] **Daniel Prusa** [1]

Abstract

In classification with a reject option, the classifier is allowed in uncertain cases to abstain from prediction. The classical cost based model of an optimal classifier with a reject option requires the cost of rejection to be defined explicitly. An alternative bounded-improvement model, avoiding the notion of the reject cost, seeks for a classifier with a guaranteed selective risk and maximal cover. We prove that both models share the same class of optimal strategies, and we provide an explicit relation between the reject cost and the target risk being the parameters of the two models. An optimal rejection strategy for both models is based on thresholding the conditional risk defined by posterior probabilities which are usually unavailable. We propose a discriminative algorithm learning an uncertainty function which preserves ordering of the input space induced by the conditional risk, and hence can be used to construct optimal rejection strategies.

1. Introduction

In classification with a reject option the classifier is allowed in uncertain cases to abstain from prediction. This setting is essential e.g. in medical diagnosis, safety critical systems and many other applications. The cost-based model of an optimal classification strategy with the reject option was proposed by Chow in his pioneering work (Chow, 1970). The goal is to minimize the expected loss equal to the cost of misclassification, if the classifier predicts, and to the reject cost, if the classifier abstains from prediction. The well known Bayes-optimal strategy rejects to predict whenever the conditional risk exceeds the reject cost. Computation of the conditional risk requires the class posterior probabilities which are usually unavailable. A common solution is the plug-in rule obtained after replacing the posterior probabilities by their estimates. A quality of the plug-in rule heavily depends on the estimated probabilities (Fumera et al., 2000; Herbei & Wegkamp, 2006). Discriminative methods instead learn the reject option classification strategy directly without the probability estimation, e.g. (Bartlett & Wegkamp, 2008; Ramaswamy & Agarwal, 2016). Many works assume that the uncertainty measure is known, e.g. obtained from responses of trained Neural Network (LeCun et al., 1990), and the task is to find only the rejection thresholds (Tortorella, 2000; Kummert et al., 2016; Fischer et al., 2016). Most existing methods, including this paper, account for aleatoric uncertainty only, while literature dealing with the epistemic uncertainty is scarce (Wang et al., 2017).

The cost-based model requires the reject cost to be defined explicitly which is difficult in some applications like e.g. medical diagnosis. An alternative bounded-improvement model proposed by (Pietraszek, 2005) avoids explicit definition of the reject cost. The rejection strategy is evaluated by two antagonistic quantities: i) the selective risk defined as the expected misclassification cost on the non-reject region and ii) the coverage corresponding to the probability mass of the non-reject region. An optimal strategy for the bounded-improvement model maximizes the coverage under the condition that the selective risk does not exceed a specified target value. In contrast to the cost-based model, it has not been formally shown what is the optimal class of strategies when the underlying model is known. The solution was proposed only for special instances of the task. (Pietraszek, 2005) proposed a method based on ROC analysis applicable when a score proportional to posterior probabilities is known and the task is to find only the optimal thresholds. (El-Yaniv & Wiener, 2010) proposed an algorithm learning the optimal strategy in the noise-free setting, i.e. when a perfect strategy with zero selective risk exists. (Geifman & El-Yaniv, 2017) shows how to equip a trained classifier with a reject option provided an uncertainty measure is known and the task is to find a rejection threshold optimal under the bounded-improvement model.

This paper describes three contributions:

1) We provide necessary and sufficient conditions for an optimal strategy of the bounded-improvement model when the underlying distribution is known. We show that an optimal

[1]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. Correspondence to: Vojtech Franc <xfrancv@cmp.felk.cvut.cz>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

rejection strategy can be constructed based on thresholding the class conditional risk, which corresponds to the optimal strategy of the cost-based model. We show that in contrast to the cost-based model, the rejection strategy cannot be arbitrary at the boundary cases. We provide an explicit relation between the parameters of the two rejection models.

2) We propose a novel loss which evaluates the quality of an uncertainty function associated to a given classifier. We prove that any minimizer of an expectation of the loss is a function which preserves ordering of the input space induced by the true conditional risk. Hence, any minimizer of the loss can be used to construct a strategy optimal for cost-based and/or bounded-improvement model.

3) We propose a discriminative algorithm which for a given classifier learns an uncertainty function by minimizing a convex surrogate of the proposed loss. We show experimentally that the learned rejection strategies outperform plug-in rules constructed from logistic-regression model and rules based on distance to the SVM decision boundary.

To our knowledge we are the first to propose a discriminative method learning an uncertainty function with a clear connection to the conditional risk being the key quantity for constructing the optimal reject-option strategies. Our method is potentially most useful when modeling the posterior distribution is difficult.

The paper organization: Section 2 characterizes optimal strategies of the bounded-improvement rejection model and relation to the cost-based model. Section 3 proposes an algorithm learning uncertainty function optimal for both models. Related works are discussed in Section 4, experiments are presented in Section 5 and a summary is given in Section 6. Proofs are referred to a supplementary material.

2. Equivalence of Cost-based and Bounded-improvement Rejection Models

Let $\mathcal{X}$ be a set of input observations and $\mathcal{Y}$ a finite set of labels. Let us assume that inputs and labels are generated by a random process with p.d.f. $p(x, y)$ defined over $\mathcal{X} \times \mathcal{Y}$. A goal in the non-reject setting is to find a *classifier* $h\colon \mathcal{X} \to \mathcal{Y}$ with a small expected risk

$$R(h) = \int_{\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x, y)\ell(y, h(x))\, dx\ ,$$

where $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a *loss* penalizing the predictions.

The expected risk can be reduced by abstaining from prediction in uncertain cases. To this end, we use a *selective classifier* [1] (h, c) composed of a classifier $h\colon \mathcal{X} \to \mathcal{Y}$ and a *selection function* $c\colon \mathcal{X} \to [0, 1]$. When applying the selective classifier to input $x \in \mathcal{X}$ it outputs

$$(h, c)(x) = \begin{cases} h(x) & \text{with probability} \quad c(x)\,, \\ \text{reject} & \text{with probability} \quad 1 - c(x)\,. \end{cases}$$

In the sequel we introduce two models of an optimal selective classifier, the cost-based and the bounded-improvement model, and we characterize their optimal strategies provided the underlying model $p(x, y)$ is known. While the solution of the former model is well known the latter is novel.

Cost-based model seeks for a selective classifier (h_B, c_B) which for a given reject cost $\varepsilon > 0$ minimizes

$$R_B(h, c) = \int_{\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x, y)\,\big(\ell(y, h(x))c(x) + (1 - c(x))\varepsilon\big)dx.$$

The well-known optimal strategy (i.e. the Bayes classifier) reads

$$h_B(x) \in \operatorname*{argmin}_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} p(y \mid x)\,\ell(y, \hat{y}) \tag{1}$$

$$c_B(x) = \begin{cases} 1 & \text{if} \quad r^*(x) < \varepsilon\,, \\ \tau & \text{if} \quad r^*(x) = \varepsilon\,, \\ 0 & \text{if} \quad r^*(x) > \varepsilon\,, \end{cases} \tag{2}$$

where $r^*(x) = \min_{\hat{y}\in\mathcal{Y}} \sum_{y\in\mathcal{Y}} p(y \mid x)\,\ell(y, \hat{y})$ is the minimal class conditional risk associated to the input x, and τ is any number from the interval $[0, 1]$. Hence there is an infinite number of optimal selection functions parameterized by τ which yield the same risk $R_B(h, c)$.

Bounded-improvement model characterizes the selective classifier by two antagonistic quantities: i) the *coverage*

$$\phi(c) = \int_{\mathcal{X}} p(x)\, c(x)\, dx$$

corresponding to the probability mass of the non-reject region and ii) the *selective risk*

$$R_S(h, c) = \frac{\int\limits_{\mathcal{X}} \sum\limits_{y\in\mathcal{Y}} p(x, y)\,\ell(y, h(x))\, c(x)\, dx}{\phi(c)}\ ,$$

defined for non-zero $\phi(c)$ as the expected classification loss on the non-reject region $\mathcal{X}_{c(x)>0}$ [2]. Given a *target risk*

[1] The *classifier with a reject option* is more frequently denoted by a single function $h'\colon \mathcal{X} \to \mathcal{Y} \cup \{\text{reject}\}$. Instead, we use the decomposition $(h, c)(x) = h'(x)$, and the terminology *selective classifier* from (El-Yaniv & Wiener, 2010) as it simplifies notation.

[2] For a function $f\colon \mathcal{X} \to \mathbb{R}$ and $a \in \mathbb{R} \cup \{\infty\}$, we define

$\mathcal{X}_{f(x)\le a} = \{x \in \mathcal{X} \,|\, f(x) \le a\}$, $\mathcal{X}_{f(x)<a} = \{x \in \mathcal{X} \,|\, f(x) < a\}$,
$\mathcal{X}_{f(x)=a} = \{x \in \mathcal{X} \,|\, f(x) = a\}$, $\mathcal{X}_{f(x)>a} = \{x \in \mathcal{X} \,|\, f(x) > a\}$,
$\mathcal{X}_{f(x)\ge a} = \{x \in \mathcal{X} \,|\, f(x) \ge a\}$.

$\lambda > 0$, the bounded-improvement model defines the optimal selective classifier (h_S, c_S) as a solution to the problem

$$\max_{h,c} \phi(c) \quad \text{s.t.} \quad R_S(h,c) \leq \lambda\,, \tag{3}$$

where we assume that both maximizers exist.

Theorem 1 *Let (h, c) be an optimal solution to (3). Then, (h_B, c), where h_B is the optimal Bayes classifier (1), is also optimal to (3).*

According to Theorem 1 the Bayes classifier h_B is also optimal for the risk-coverage task (3) which is not surprising. Note the Bayes classifier is not a unique solution to (3) because the predictions on the reject region $\mathcal{X}_{c(x)=0}$ do not count to the selective risk and hence they can be arbitrary.

Theorem 1 allows to solve the bounded-improvement task (3) in two consecutive steps: First, set h to be the Bayes classifier h_B or to its best approximation, e.g. learned by a discriminative method of choice. Second, when h is fixed, the optimal selection function c^* is obtained by solving the task (3) only w.r.t c which boils down to

$$\max_{c\in[0,1]^{\mathcal{X}}} \int_{\mathcal{X}} p(x)c(x)dx \;\; \text{s.t.} \;\; \int_{\mathcal{X}} p(x)c(x)\overline{r}(x)dx \leq 0 \tag{4}$$

where $\overline{r}(x) = r(x) - \lambda$ measures how much the conditional risk of the prediction $h(x)$, defined as

$$r(x) = \sum_{y\in\mathcal{Y}} p(y \mid x)\, \ell(y, h(x))\,, \tag{5}$$

exceeds the target risk λ. It is seen that for $h = h_B$ solving (4) yields c^* which is a solution to the task (3), i.e. $c^* = c_S$. Indeed, using the definition of $R_S(h,c)$ we can rewrite the constraint of the task (3) as

$$\int_{\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y \mid x)\ell(y, h(x))c(x)\, dx \leq \lambda \int_{\mathcal{X}} p(x)c(x)\, dx$$

which after a simple rearrangement becomes the constraint of the task (4), and objectives of both tasks are the same.

Theorem 2 *A selection function $c^* : \mathcal{X} \to [0,1]$ is an optimal solution to (4) if and only if it holds*

$$\int_{\mathcal{X}_{\overline{r}(x)<b}} p(x)c^*(x)dx = \int_{\mathcal{X}_{\overline{r}(x)<b}} p(x)dx, \tag{6}$$

$$\int_{\mathcal{X}_{\overline{r}(x)=b}} p(x)c^*(x)dx = \begin{cases} -\frac{\rho(\mathcal{X}_{\overline{r}(x)<b})}{b} & \text{if} \quad b > 0\,, \\ \int_{\mathcal{X}_{\overline{r}(x)=0}} p(x)dx & \text{if} \quad b = 0\,, \end{cases} \tag{7}$$

$$\int_{\mathcal{X}_{\overline{r}(x)>b}} p(x)c^*(x)dx = 0\,, \tag{8}$$

where $\rho(\mathcal{X}') = \int_{\mathcal{X}'} p(x)\overline{r}(x)\, dx$ is the expectation of $\overline{r}(x)$ restricted to inputs in $\mathcal{X}'$, and

$$b = \sup\{a \mid \rho(\mathcal{X}_{\overline{r}(x)\leq a}) \leq 0\} \geq 0\,. \tag{9}$$

Theorem 2 defines behaviour of optimal selection function on the partition of the input space $\mathcal{X}$ into three regions $\mathcal{X}_{\overline{r}(x)<b}$, $\mathcal{X}_{\overline{r}(x)=b}$ and $\mathcal{X}_{\overline{r}(x)>b}$. The condition (6) says that the conditional expectation $\mathbb{E}_{x\sim p(x)}[c^*(x) - 1 \mid \overline{r}(x) < b]$ is zero, which is satisfied if for $\overline{r}(x) < b$ we use $c^*(x) = 1$. Or equivalently, by using the identity $\overline{r}(x) = r(x) - \lambda$, we set $c^*(x) = 1$ whenever $r(x) < \gamma$ with the threshold $\gamma = b + \lambda$. Analogically, setting $c^*(x) = 0$ for $r(x) > \gamma$ satisfies the condition (8). Finally, if we opt for a constant selecting function $c^*(x) = \tau$ in the boundary region $\mathcal{X}_{\overline{r}(x)=b}$, then the condition (7) implies $\tau = -\frac{\rho(\mathcal{X}_{r(x)<\gamma})}{b\cdot\rho_0}$ if $b > 0$, where $\rho_0 = \int_{\mathcal{X}_{\overline{r}(x)=b}} p(x)\, dx$, and $\tau = 1$ if $b = 0$. Using $\mathcal{X}_{\overline{r}(x)<b} = \mathcal{X}_{r(x)<\gamma}$ and $b \cdot \rho_0 = \rho(\mathcal{X}_{\overline{r}(x)=b}) = \rho(\mathcal{X}_{r(x)=\gamma})$, we derive

$$\tau = \begin{cases} 1 & \text{if} \quad \rho(\mathcal{X}_{r(x)=\gamma}) = 0\,, \\ -\frac{\rho(\mathcal{X}_{r(x)<\gamma})}{\rho(\mathcal{X}_{r(x)=\gamma})} & \text{if} \quad \rho(\mathcal{X}_{r(x)=\gamma}) > 0\,. \end{cases} \tag{10}$$

Corollary 1 *Let $r\colon \mathcal{X} \to \mathbb{R}$ be the conditional risk (5), τ the acceptance probability given by (10) and $\gamma = b + \lambda$ the rejection threshold given by the target-risk λ and b computed by (9). Then the selection function*

$$c^*(x) = \begin{cases} 1 & \textit{if} \quad r(x) < \gamma\,, \\ \tau & \textit{if} \quad r(x) = \gamma\,, \\ 0 & \textit{if} \quad r(x) < \gamma\,, \end{cases} \tag{11}$$

satisfies the optimality condition of Theorem 2.

It is seen that (11) coincides with the Bayes-optimal selection function (2) of the cost-based model. In case of the cost-based model, the threshold γ corresponds to the reject cost ε while the acceptance probability τ can be arbitrarily without affecting the risk $R_B(h,c)$. In contrast, the optimal threshold $\gamma = b + \lambda$ of the bounded-improvement model is defined implicitly by choosing the target risk λ. In addition, the acceptance probability τ for the boundary cases cannot be arbitrary in general. In practice, however, the impact of τ is negligible because ρ_0 is usually small or even zero as can be expect for continuous $p(x)$ when the boundary region $\mathcal{X}_{r(x)=\gamma}$ has probability measure zero. On the other hand, setting the threshold γ is crucial. Fortunately, (Geifman & El-Yaniv, 2017) shows how to find γ from a finite training set such that it is optimal solution of the bounded-improvement model in PAC sense. The remaining issue is thus the conditional risk $r(x)$ because its computation requires the posterior probabilities $p(y \mid x)$. A common solution is the *plug-in rule* obtained after replacing $p(y \mid x)$ in (5) by $\hat{p}(y \mid x)$ estimated from examples. In the next section, we propose an alternative approach which completely avoids the probability estimation.

3. Discriminative Learning of Uncertainty

Assume a classifier $h\colon \mathcal{X} \to \mathcal{Y}$ has been fixed and we want to endow it with a selection function $c\colon \mathcal{X} \to [0,1]$. We have shown that regardless the rejection model, the optimal selection function is based on thresholding the conditional risk $r(x)$ defined by (5). Hence, it is clear that $r(x)$ can be replaced by any uncertainty function $s\colon \mathcal{X} \to \mathbb{R}$ which preserves ordering of inputs induced by the conditional risk $r(x)$ [3]. In this section, we show how to learn such uncertainty functions from examples.

3.1. Order Enforcing Loss Function

Let $\mathcal{T}_n = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, n\}$ be a set of inputs and labels generated from n i.i.d. random variables with distribution $p(x,y)$. We define a loss function $\Delta\colon \mathbb{R}^n \times \mathcal{X}^n \times \mathcal{Y}^n \to \mathbb{R}_+$ as

$$\Delta(s, \mathcal{T}_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \ell(y_i, h(x_i)) [\![s(x_i) \leq s(x_j)]\!] \,, \tag{12}$$

which has the following interpretation. Let us order the examples $\mathcal{T}_n$ according to $s(x)$ so that $s(x_{\pi(1)}) \leq s(x_{\pi(2)}) \leq \cdots \leq s(x_{\pi(n)})$, where $\pi\colon \{1, \ldots, n\} \to \{1, \ldots, n\}$ is the permutation defining the order. Let $\hat{R}(i,s) = \frac{1}{n} \sum_{j=1}^{i} \ell(y_{\pi_j}, h(x_{\pi_j})) [\![s(x_{\pi_j}) \leq s(x_{\pi_i})]\!]$ be the empirical risk of the classifier $h(x)$ computed on the examples with uncertainty *not higher than* the uncertainty of the i-th example. Then, the loss $\Delta(s, \mathcal{T}_n)$ can be seen as the area under the curve $\mathcal{C} = \{(\hat{R}(i,s), \frac{i}{n}) \mid i = 1, \ldots, n\}$. The curve $\mathcal{C}$ summarizes performance of the selective classifier on $\mathcal{T}_n$ when it rejects to predict based on thresholding the uncertainty $s(x)$. It is intuitively clear that the area is minimized by such $s(x)$ which orders the examples as if they were ordered according to the loss $\ell(y_i, h(x_i))$. Hence, by minimizing the expectation of $\Delta(s, \mathcal{T}_n)$ we should get a function preserving the ordering induced by the conditional risk. In the sequel we will support the intuition by theorems.

Let us define the expectation of the loss $\Delta(s, \mathcal{T}_n)$ as

$$\begin{aligned}
E(s) &= \int_{\mathcal{X}^n} \sum_{\boldsymbol{y} \in \mathcal{Y}^n} \prod_{i=1}^{n} p(x_i, y_i) \Delta(s, \mathcal{T}_n) d\boldsymbol{x} \\
&= \frac{1}{n^2} \int_{\mathcal{X}^n} \prod_{i=1}^{n} p(x_i) \sum_{i=1}^{n} \sum_{j=1}^{n} r(x_i) [\![s(x_i) \leq s(x_j)]\!] \, d\boldsymbol{x} \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int_{\mathcal{X}} \int_{\mathcal{X}} p(x) p(z) r(x) [\![s(x) \leq s(z)]\!] dz \, dx \\
&= \int_{\mathcal{X}} p(x) \, r(x) \left(\int_{\mathcal{X}} p(z) \, [\![s(x) \leq s(z)]\!] dz \right) dx \,,
\end{aligned}$$

its minimizers are described by the following theorem [4].

Theorem 3 *A function $s^* : \mathcal{X} \to \mathbb{R}$ is an optimal solution to $\min_{s:\mathcal{X}\to\mathbb{R}} E(s)$ if and only if*

$$\int_{\mathcal{X}} \int_{\substack{z \neq x \\ s^*(z) = s^*(x)}} \max\{r(x), r(z)\} p(x) p(z) dz \, dx = 0 \,, \text{ and} \tag{13}$$

$$\int_{\mathcal{X}} \int_{\substack{r(z) < r(x) \\ s^*(z) > s^*(x)}} (r(x) - r(z)) \, p(x) p(z) dz \, dx = 0 \,. \tag{14}$$

The conditions (13) and (14) imply that the conditional expectations $\mathbb{E}_{x,z\sim p(x)}[\max\{r(x), r(z)\} \mid z \neq x \wedge s^*(x) = s^*(z)]$ and $\mathbb{E}_{x,z\sim p(x)}[r(x) - r(z) \mid r(z) < r(x) \wedge s^*(z) > s^*(x)]$ are both zero. If combined it further implies that a subset of input space $\mathcal{X}' = \{(x,z) \in \mathcal{X} \times \mathcal{X} \mid r(z) < r(x) \wedge s^*(z) > s^*(x)\}$ on which the order is violated has probability measure 0. In other words the optimal $s^*(x)$ preserves the ordering induced by $r(x)$ *almost surely*.

Corollary 2 *Any function $s : \mathcal{X} \to \mathbb{R}$ fulfilling*

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X} : x \neq x' \Rightarrow s(x) \neq s(x'), \text{ and} \tag{15}$$

$$\forall (x, x') \in \mathcal{X} \times \mathcal{X} : r(x) < r(x') \Rightarrow s(x) < s(x') \tag{16}$$

satisfies the optimality conditions of Theorem 3.

Corollary 2 suggests how to construct the optimal $s^*(x)$ if we know the conditional risk $r(x)$. More importantly, the following corollary states that the optimal $s^*(x)$ can be used instead of $r(x)$ to construct the optimal selection function.

Corollary 3 *Let $s^* \in \operatorname{argmin}_{s:\mathcal{X}\to\mathbb{R}} E(s)$. The selection function $c^* : \mathcal{X} \to [0,1]$ defined by*

$$c^*(x) = \begin{cases} 1 & \text{if} \quad s^*(x) < \gamma \,, \\ \tau & \text{if} \quad s^*(x) = \gamma \,, \\ 0 & \text{if} \quad s^*(x) > \gamma \,, \end{cases} \tag{17}$$

$$\text{where} \quad \gamma = \sup \{a \mid \rho(\mathcal{X}_{s^*(x) \leq a}) \leq 0\} \,, \tag{18}$$

$$\text{and} \quad \tau = \begin{cases} 1 & \text{if} \quad \rho(\mathcal{X}_{s^*(x)=\gamma}) = 0 \,, \\ -\frac{\rho(\mathcal{X}_{s^*(x)<\gamma})}{\rho(\mathcal{X}_{s^*(x)=\gamma})} & \text{if} \quad \rho(\mathcal{X}_{s^*(x)=\gamma}) > 0 \,, \end{cases} \tag{19}$$

fulfills conditions (6), (7) and (8) of Theorem 2, therefore it is an optimal solution to (4).

3.2. Algorithm

The expectation $E(s)$ cannot be minimized directly because $p(x,y)$ is unknown and the loss $\Delta(s, \mathcal{T}_n)$ is hard to optimize due to the step function $[\![0 \leq t]\!]$ in its definition. We

[3] $f\colon \mathcal{X} \to \mathbb{R}$ preserves ordering induced by $g\colon \mathcal{X} \to \mathbb{R}$ iff $\forall (x, x') \in \mathcal{X} \times \mathcal{X}, g(x) < g(x') \Rightarrow f(x) < f(x')$.

[4] $\int_{\mathcal{X}} \int_{\substack{z \neq x \\ s^*(z) = s^*(x)}} f(x,z) dz \, dx$ stands for $\int_{\mathcal{X}} \int_{\mathcal{X}'} f(x,z) dz \, dx$ where $\mathcal{X}' = \{z \in \mathcal{X} \mid z \neq x \wedge s^*(z) = s^*(x)\}$, etc.

describe two standard techniques to minimize $E(s)$ approximately given a training set $\mathcal{T}_m = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} \mid i = 1, \ldots, m\}$ drawn i.i.d. from $p(x, y)$. We assume that uncertainty function $s_{\boldsymbol{\theta}} \colon \mathcal{X} \to \mathbb{R}$ is known up to parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ which need to be learned from $\mathcal{T}_m$.

Stochastic Gradient Approximation Starting from initial $\boldsymbol{\theta}_0 \in \mathbb{R}^n$, the new values are computed in iterative fashion by $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + k_t \, \boldsymbol{g}_t$, where k_t is an appropriate learning rate and $\boldsymbol{g}_t$ is a sub-gradient evaluated at $\boldsymbol{\theta}_t$ of the proxy-loss

$$\hat{\Delta}(\boldsymbol{\theta}, \mathcal{T}_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \ell(y_i, h(x_i)) \, \omega\big(s_{\boldsymbol{\theta}}(x_j) - s_{\boldsymbol{\theta}}(x_i)\big) \tag{20}$$

where $\omega(t) = \max\{0, 1 + t\}$ is a convex upper-bound of $[\![0 \leq t]\!]$ used in definition of the original loss $\Delta(s_{\boldsymbol{\theta}}, \mathcal{T}_n)$. The set $\mathcal{T}_n$ denotes a mini-batch selected randomly from $\mathcal{T}_m$. It is seen that the true objective $E(s_{\boldsymbol{\theta}})$ is upper bounded by the expectation (w.r.t. randomly generated $\mathcal{T}_n$) of $\hat{\Delta}(\boldsymbol{\theta}, \mathcal{T}_n)$ which we use as our proxy, and which can be optimized by SGD (Kushner & Yin, 2003) or similar algorithms.

Regularized Empirical Risk Approximation We approximate $E(s)$ by

$$F(\boldsymbol{\theta}, \mathcal{T}_m) = \frac{C}{2} \|\boldsymbol{\theta}\|^2 + \frac{1}{P} \sum_{k=1}^{P} \hat{\Delta}(\boldsymbol{\theta}, \mathcal{T}^k) \tag{21}$$

where $\mathcal{T}^1 \cup \mathcal{T}^2 \cup \cdots \cup \mathcal{T}^P$ is randomly generated partition of the training set $\mathcal{T}_m$ into P approximately equally sized batches, and $C > 0$ is a chosen regularization constant. It is seen that the expectation (w.r.t. randomly generated $\mathcal{T}_m$) of $F(\boldsymbol{\theta}, \mathcal{T}_m)$ upper bounds $E(s)$. The objective $F(\boldsymbol{\theta}, \mathcal{T}_m)$ is convex in $\boldsymbol{\theta}$ for uncertainty functions $s_{\boldsymbol{\theta}}(x) = \langle \boldsymbol{\theta}, \boldsymbol{\psi}(x) \rangle$ linear in $\boldsymbol{\theta}$ and then any convex solver can be used.

4. Related Works

The cost-based rejection model was proposed in (Chow, 1970) who also provides the optimal strategy in case the distribution is known, analyzes the error-reject trade-off, and proves basic properties of the error-rate and the reject-rate, e.g. that both functions are monotone w.r.t. the reject cost. The original paper considers the risk with 0/1-loss only. The model with arbitrary prediction costs was analyzed e.g. in (Tortorella, 2000; Schlesinger & Hlaváč, 2002).

The bounded-improvement model was proposed in (Pietraszek, 2005). He assumed that the classifier score proportional to the posterior probabilities is known and the task is to find only the optimal thresholds, which is done numerically based on ROC analysis. The original formulation assumes two classes and 0/1-loss. In our paper we consider a straightforward generalization of the bounded-improvement model, defined by task (3), which allows arbitrary number of classes, arbitrary loss and puts no constraint on the class of optimal strategies. We show how to construct the optimal strategy provided the underlaying distribution $p(x, y)$ is known.

Learning of a selective classifier optimal for the bounded-improvement model was discussed in (El-Yaniv & Wiener, 2010). Their method requires a noisy-free scenario, i.e. there must exist a selective classifier with $R_S(h, c) = 0$. They also provide a characterization of the lower and upper bound of the risk-coverage curves in PAC setting. (Geifman & El-Yaniv, 2017) proposed a method to find a threshold optimal in PAC sense provided the uncertainty function $s(x)$ is known, which in their work is constructed from an output of a trained Neural Network. We complement their work by a method learning the uncertainty function $s(x)$.

A common approach to construct a rejection strategy is based on *plug-in* conditional risk $\hat{r}(x)$ obtained after replacing $p(y \mid x)$ in (5) by their estimate $\hat{p}(y \mid x)$. (Fumera et al., 2000) shows that the plug-in Bayes-optimal strategy is not optimal if the estimates $\hat{p}(y \mid x)$ are affected by errors, in which case they propose to use class-related thresholds instead of a single global threshold. Other methods trying to improve the plug-in strategy by tuning multiple thresholds were proposed in (Kummert et al., 2016; Fischer et al., 2016). The statistically consistency of the plug-in reject rules is discussed in (Herbei & Wegkamp, 2006). We use the plug-in rule on top of logistic-regression model (McCullagh & Nelder, 1989) as a baseline in our experiments.

Most prediction models come with at least an ordinal measure of uncertainty which can be used to construct the reject strategies. E.g. (LeCun et al., 1990) proposed a reject strategy for a Neural Network classifier based on thresholding either i) the output of the maximally activated unit of the last layer or ii) a difference between the maximal and runner upper output units. In case of Support Vector Machine classifiers (Vapnik, 1998) the trained linear score, proportional to the distance between the input and the decision hyperplane, is often used as the uncertainty measure (Fumera & Roli, 2002). We consider this approach as a baseline in our experiments. Other solutions involve fitting a probabilistic model to the SVM outputs (Platt, 2000; Wu et al., 2004).

5. Experiments

We experimented with two methods to learn the classifier $h(x)$ and two baselines to compute the uncertainty measure: i) Logistic-Regression estimating $\hat{p}(y \mid x)$ so that the plug-in Bayes classifier and the conditional risk can be computed, ii) the SVMs, representing discriminative methods without probabilistic output, in which case the classification

dataset	classes	features	examples
AVILA	12	10	20,867
COVTYPE	7	54	581,012
CODRNA	2	8	331,152
IJCNN	2	22	49,990
LETTER	26	16	20,000
PENDIGIT	10	16	10,992
PHISHING	2	68	11,055
SATTELITE	6	36	6,435
SENSORLESS	11	48	58,509
SHUTTLE	7	9	58,000

Table 1. Summary of benchmark classification problems.

score with the maximal response was used as an uncertainty measure. Then, on top of the LR and SVM classifiers we learned the uncertainty measure by minimizing the proposed loss. We implemented both variants based on the stochastic gradient approximation and the regularized empirical risk minimization. The methods were evaluated on 10 classification problems with different sizes, number of features and number of classes. The goal was to minimize the classification error, hence we used the 0/1-loss $\ell(y, y') = [\![y \neq y']\!]$.

Logistic-regression (LR) learns parameters $\boldsymbol{\theta} = ((\boldsymbol{w}_y, b_y) \in \mathbb{R}^d \times \mathbb{R} \mid y \in \mathcal{Y})$ of the posterior probabilities $\hat{p}_{\boldsymbol{\theta}}(y \mid x) \approx \exp(\langle \boldsymbol{w}_y, \boldsymbol{x} \rangle + b_y)$ by maximizing the regularized log-likelihood $L(\boldsymbol{\theta}) = \frac{C}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{m}\sum_{i=1}^m \log\big(\hat{p}_{\boldsymbol{\theta}}(y_i \,|\, \boldsymbol{x}_i)\big)$. The optimal C was selected from $\{1, 10, 100, 1000\}$ based on the validation classification error. After learning $\boldsymbol{\theta}$, we used the plug-in Bayes classifier $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \operatorname{argmax}_{y\in\mathcal{Y}} \hat{p}_{\boldsymbol{\theta}}(y \mid \boldsymbol{x})$ and the *plug-in* class conditional risk $\hat{r}(\boldsymbol{x}) = 1 - \hat{p}_{\boldsymbol{\theta}}(h(\boldsymbol{x}) \,|\, \boldsymbol{x})$ as the baseline uncertainty measure. Note that both plug-in rules are Bayes-optimal for the 0/1-loss.

Support Vector Machines (SVM) learn parameters $\boldsymbol{\theta} = ((\boldsymbol{w}_y, b_y) \in \mathbb{R}^d \times \mathbb{R} \mid y \in \mathcal{Y})$ of the linear classifier $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \operatorname{argmax}_{y\in\mathcal{Y}} \langle \boldsymbol{w}_y, \boldsymbol{x} \rangle + b_y$ by minimizing $F(\boldsymbol{\theta}) = \frac{C}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{m}\sum_{i=1}^m \max_{y\in\mathcal{Y}} \big([\![y \neq y_i]\!] + \langle \boldsymbol{w}_y - \boldsymbol{w}_{y_i}, \boldsymbol{x}_i \rangle\big)$. As the baseline uncertainty measure we use the *maximal score* $s(\boldsymbol{x}) = \max_{y\in\mathcal{Y}} \langle \boldsymbol{w}_y, \boldsymbol{x} \rangle + b_y$ proportional to the distance between input x and the decision hyper-plane. For binary case $|\mathcal{Y}| = 2$, we used $\boldsymbol{\theta} = (\boldsymbol{w}, b)$, $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b)$, $F(\boldsymbol{\theta}) = \frac{C}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{m}\sum_{i=1}^m \max\{0, 1 - y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b)\}$ and $s(\boldsymbol{x}) = |\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b|$. The optimal C was selected in the same way as for LR.

Proposed method for uncertainty learning We parameterized the uncertainty function as

$$s_{\boldsymbol{\theta}}(\boldsymbol{x}) = \langle \boldsymbol{w}_{h(\boldsymbol{x})}, \boldsymbol{\psi}(\boldsymbol{x}) \rangle + b_y \,, \tag{22}$$

where $\boldsymbol{\psi} \colon \mathbb{R}^d \to \mathbb{R}^q$ are features extracted from the vectorial inputs $\boldsymbol{x} \in \mathbb{R}^d$, and $\boldsymbol{W} = ((\boldsymbol{w}_y, b_y) \in \mathbb{R}^q \times \mathbb{R} \mid y \in \mathcal{Y})$ are unknown parameters. We experimented with three different feature maps: i) linear features $\boldsymbol{\psi}(\boldsymbol{x}) = \boldsymbol{x}$, ii) quadratic features $\boldsymbol{\psi}(\boldsymbol{x}) = (x_i \cdot x_j \mid (i,j) \in \{1, \ldots, d\}^2 \wedge j \leq i)$ and iii) features extracted by multi-layer perceptron trained from examples. For linear and quadratic features, $\boldsymbol{\psi}$ is fixed and the unknown parameters $\boldsymbol{\theta} = \boldsymbol{W}$ were learned by minimizing the convex objective function (21) using the Bundle Method for Risk Minimization (Choon et al., 2010). The number of batches was $P = 5$, and the regularization constant C was selected from $\{1, 10, 100, 1000\}$ based on the proposed loss (12) evaluated on the validation set. Finally, we used MLP with 1, 5, or 10 layers (the optimal number was selected based on the validation set) each having the same number of neurons as the input dimension d. The ReLU was used as the transfer function. The unknown parameters $\boldsymbol{\theta} = (\boldsymbol{\psi}, \boldsymbol{W})$ comprised of the linear filters of fully-connected layers $\boldsymbol{\psi}$ and parameters $\boldsymbol{W}$ of the rule (22) put as the last layer. The parameters $\boldsymbol{\theta}$ were learned by ADAM (Kingma & Ba, 2015) optimizing the convex loss function (20). To speed up convergence, we used the batch-normalization (Ioffe & C. Szegedy, 2015) placed after each fully-connected layer.

5.1. Datasets

We selected 10 classification problems from UCI repository (Dua & Taniskidou, 2017) and libSVM datasets (Chang & C.J.Lin, 2011). The datasets are summarized in Table 1. We chose the datasets with sufficiently large number of examples, as we need to learn both the classifier and the uncertainty, and with a moderate input dimension d because of using $q = d(d+1)/2$ explicitly computed quadratic features. Each dataset was randomly split 5 times into 5 subsets, Trn1/Val1/Trn2/Val2/Tst, in ratio $30/10/30/10/20$ (up to COVTYPE with ratio $28/20/2/20/30$). The subsets Trn1/Val1 and Trn2/Val2 were used for learning and model selection of the classifier and the uncertainty function, respectively. All features were normalized to have zero mean and unit variance. The normalization coefficients were estimated using only the Trn1 and Trn2 subsets, respectively. The Tst subset was used solely to compute the test metrics.

5.2. Evaluation Protocol

Selective classifiers are evaluated based on Risk-Coverage curve $\{(\hat{R}_S(i,s), \frac{i}{n}) \mid i = 1, \ldots, n\}$ where $\hat{R}_S(i,s)$ is an estimate of $R_S(h,c)$ computed on test examples by

$$\hat{R}_S(i,s) = \frac{1}{i}\sum_{j=1}^{i} \ell(y_{\pi(j)}, h(x_{\pi(j)}))\,,$$

$\pi(j)$ is the permutation obtained by sorting the examples according to $s(x)$ in ascending order. We describe the performance by the selective risk at coverage 90%, defined as $R@90 = \hat{R}_S(i,s)$ where $i = \operatorname{round}(0.9\,n)$, and by the Area Under risk-coverage Curve $AUC(s) =$

$\frac{1}{n}\sum_{i=1}^{n}\hat{R}_S(i,s)$. For each metric we report averages and standard deviations computed over the 5 random splits. In figures we show the mean curves.

5.3. Results

Figure 1 and Table 2 summarize results for the two baseline selective classifiers, LR+plugin and SVM+maxscore, and the selective classifiers which use uncertainty function learned by the proposed method, LR+learn(X) and SVM+learn(X), where X stands for the used feature map. The table reports results only for the best features while the figure shows all models. We can conclude that:

1) $R@100$ is always higher than $R@90$ which shows that all selective classifiers when accepting only 90% of inputs reduce the classification error of the non-rejecting classifiers.

2) The selective classifiers with uncertainty learned by the proposed method outperform the baselines both in terms of AUC and $R@90$ consistently *on all datasets*. Up to a few cases the improvement is significant.

3) LR and SVM classifiers have statistically similar error $R@100$ on 7 datasets while on 3 datasets (letter, shuttle, sensorless) SVM performs better. However, on the same 3 datasets the baseline selective classifier SVM+maxscore is worse than the baseline LR+plugin in terms of $R@90$, probably due to non-probabilistic output used as uncertainty by SVMs. At the same time, when the learned uncertainty is used the SVM-based selective classifier becomes better.

4) In 6 versus 4 cases the uncertainty functions on top of the learned MLP features outperform the hand-crafted quadratic features. While the linear features were never optimal they perform in most cases better than non-learned baselines.

6. Conclusions

We proved that the cost-based and bounded-improvement rejection models share the same optimal strategies. In both cases an optimal reject strategy can be constructed by thresholding the conditional risk. We provided formulas relating parameters of the two models. We proposed a new loss function whose minimizer is an uncertainty function which preserves ordering induced by the conditional risk, and hence can be used to construct an optimal reject strategy for both models. We experimentally showed that Logistic-Regression and SVM-based selective classifiers with uncertainty function learned by minimizing a convex-surrogate of the proposed loss outperform commonly used reject rules.

Open questions involve e.g. i) analysis of the statistical consistency of the proposed convex proxy-loss (20) and ii) investigating the possibility to simultaneously learn both the uncertainty and the prediction rule instead of using the data demanding two-stage approach.

	classifier	selection function	AUC ×100	R@90 ×100	R@100 ×100
AVILA	LR	plugin	27.2 ±0.6	40.9 ±0.5	43.7 ±0.4
	LR	learn(M)	17.3 ±0.4	38.0 ±0.4	
	SVM	maxscore	31.7 ±0.8	41.0 ±0.7	43.3 ±0.7
	SVM	learn(M)	16.9 ±0.7	37.4 ±0.8	
CODRNA	LR	plugin	0.9 ±0.0	2.0 ±0.0	4.8 ±0.1
	LR	learn(M)	0.4 ±0.0	1.3 ±0.1	
	SVM	maxscore	0.9 ±0.1	2.0 ±0.1	4.8 ±0.1
	SVM	learn(M)	0.4 ±0.0	1.3 ±0.1	
COVTYPE	LR	plugin	16.5 ±0.1	25.1 ±0.2	27.6 ±0.2
	LR	learn(M)	10.0 ±0.1	21.6 ±0.2	
	SVM	maxscore	25.7 ±0.8	26.8 ±0.1	27.4 ±0.1
	SVM	learn(M)	9.8 ±0.1	21.5 ±0.1	
IJCNN	LR	plugin	1.3 ±0.0	3.8 ±0.1	7.5 ±0.1
	LR	learn(M)	0.3 ±0.0	0.4 ±0.1	
	SVM	maxscore	1.4 ±0.0	3.7 ±0.2	7.6 ±0.2
	SVM	learn(M)	0.4 ±0.0	0.4 ±0.1	
LETTER	LR	plugin	7.4 ±0.4	18.3 ±0.6	23.3 ±0.6
	LR	learn(Q)	4.1 ±0.1	15.4 ±0.6	
	SVM	maxscore	10.2 ±0.2	19.1 ±0.4	22.1 ±0.7
	SVM	learn(Q)	3.9 ±0.3	14.0 ±0.8	
PENDIGI	LR	plugin	0.7 ±0.0	1.9 ±0.1	5.3 ±0.4
	LR	learn(Q)	0.7 ±0.1	0.8 ±0.2	
	SVM	maxscore	2.8 ±0.4	3.9 ±0.4	4.9 ±0.6
	SVM	learn(Q)	0.8 ±0.1	0.7 ±0.2	
PHISHIN	LR	plugin	0.8 ±0.1	2.9 ±0.3	6.3 ±0.4
	LR	learn(Q)	0.8 ±0.3	1.8 ±0.4	
	SVM	maxscore	0.8 ±0.1	3.0 ±0.3	6.4 ±0.4
	SVM	learn(Q)	0.7 ±0.2	1.7 ±0.3	
SHUTTLE	LR	plugin	0.6 ±0.1	1.0 ±0.1	3.4 ±0.2
	LR	learn(M)	0.1 ±0.1	0.0 ±0.0	
	SVM	maxscore	1.3 ±0.5	1.5 ±0.3	2.0 ±0.1
	SVM	learn(M)	0.1 ±0.0	0.0 ±0.0	
SENSORL	LR	plugin	2.0 ±0.1	4.9 ±0.3	8.2 ±0.4
	LR	learn(M)	0.4 ±0.0	0.6 ±0.2	
	SVM	maxscore	3.7 ±0.2	6.9 ±0.3	6.9 ±0.2
	SVM	learn(M)	0.4 ±0.0	0.5 ±0.1	
SATTELI	LR	plugin	3.8 ±0.3	11.1 ±0.3	15.1 ±0.5
	LR	learn(Q)	3.2 ±0.3	9.8 ±0.9	
	SVM	maxscore	4.8 ±0.6	11.4 ±0.5	15.4 ±0.4
	SVM	learn(Q)	3.3 ±0.2	10.2 ±0.6	

Table 2. Area Under risk-coverage Curve (AUC), selective risk at coverage 90% and 100% are shown for selective classifiers built from the non-rejection LR or SVM models. The baseline selective function for LR model uses the plug-in conditional risk and the max-score for SVM. Below each baseline we report results for the selective classifiers with learned confidence measure. We report results only for model with the best features ((L)inear, (Q)uadratic and (M)ulti-layer perceptron) selected based on the validation set.

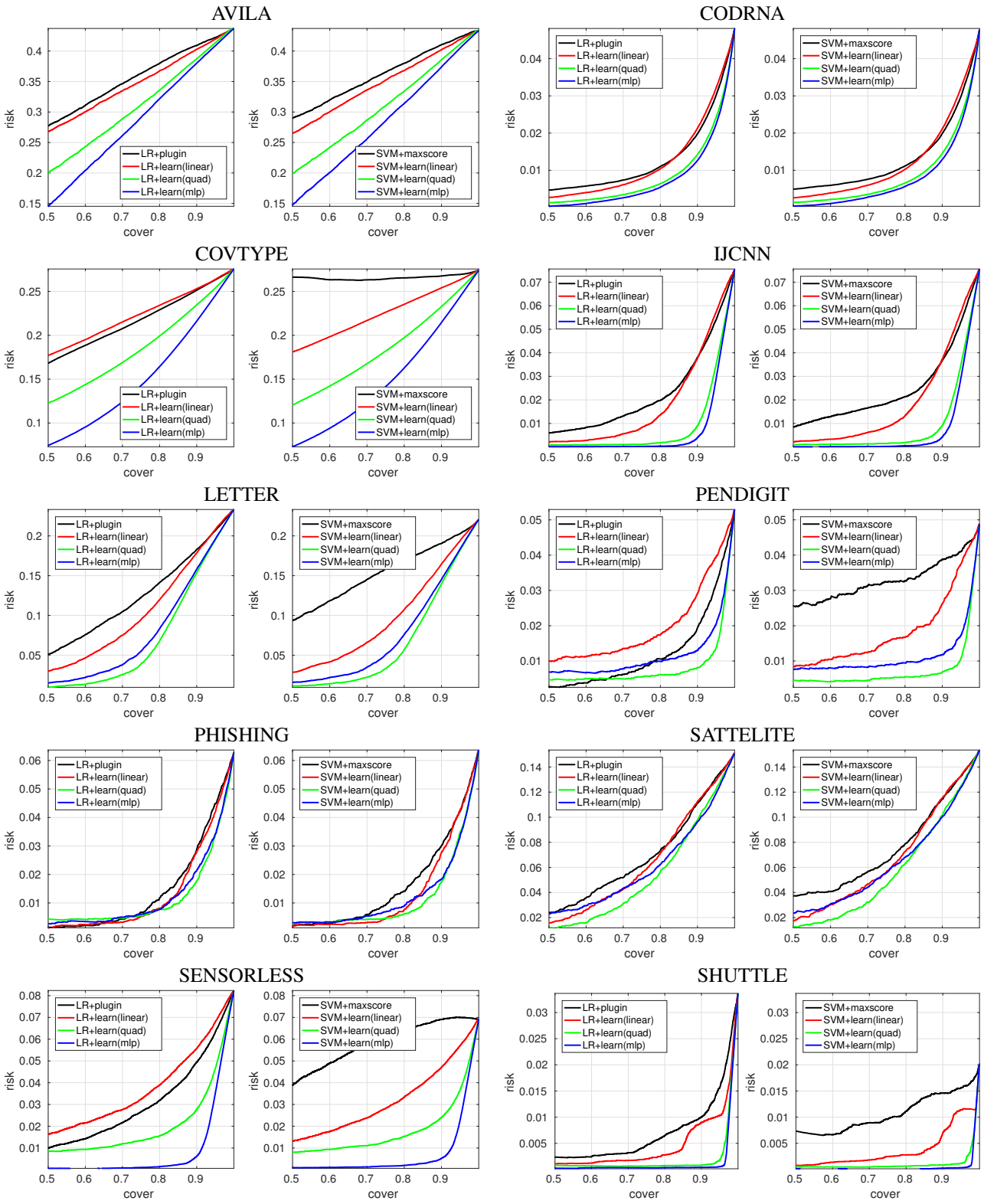

Figure 1. The risk-coverage curves for different selective classifiers evaluated on 10 datasets. X-axis is the coverage and y-axis is the selective risk corresponding to the misclassification error on the non-reject area. The classifier is either LR (left sub-figure) or SVM (right sub-figure). The curves for baseline selective functions shown in black correspond to the the plug-in conditional risk for LR and to the max-score for SVM. Colored curves correspond to selective functions learned by the proposed algorithm on top of linear features (red), quadratic features (green) and features extracted by multi-layer perceptron (blue).

Acknowledgments

We thank reviewers for their useful comments. The research was supported by the Czech Science Foundation project GACR GA19-21198S and OP VVV project CZ.02.1.01\0.0\0.0\16_019\0000765 Research Center for Informatics.

References

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

Chang, C. and C.J.Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Choon, T., Vishwanthan, S., Smola, A., and Quoc, V. L. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010. ISSN 1532-4435.

Chow, C. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theor.*, 16(1):41–46, 1970.

Dua, D. and Taniskidou, E. K. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 2010.

Fischer, L., Hammer, B., and Wersing, H. Optimal local rejection for classifiers. *Neurocomputing*, 214:445–457, 2016.

Fumera, G. and Roli, F. Support vector machines with embedded reject option. In *First International Workshop on Patter Recognition with Support Vector Machines*, 2002.

Fumera, G., Roli, F., and Giacinto, G. Multiple reject thresholds for improving classification reliability. In *Joint IAPR Internatinal Workshop on Advances in Pattern Recognition and*, pp. 863–871, 2000.

Geifman, Y. and El-Yaniv, R. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30*, pp. 4878–4887, 2017.

Herbei, R. and Wegkamp, M. Classification with reject option. *Can. J. Stat*, 34(4):709–721, 2006.

Ioffe, S. and C. Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on International Conference on Machine Learning*, pp. 448–456, 2015.

Kingma, D. and Ba, L. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kummert, J., Paassen, B., Jensen, J., Göpfert, C., and Hammer, B. Local reject option for deterministic multi-class SVM. In *Artificial Neural Networks and Machine Learning – ICANN*, 2016.

Kushner, H. and Yin, G. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jakel, L. Handwritten digit recognition with a back-propagation networks. In *Neural Information Processing Systems*, 1990.

McCullagh, P. and Nelder, J. *Generalized Linear Models*. Chapman and Hall, 1989.

Pietraszek, T. Optimizing abstaining classifiers using ROC analysis. In *International Conference on Machine Learning*, 2005.

Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola et al. (ed.), *Advances in Large Margin Classifiers*. Cambridge, MA, 2000.

Ramaswamy, H. and Agarwal, S. Convex calibration dimension for multiclass loss matrices. *Journal of Machine Learning Research*, 17:1–45, 2016.

Schlesinger, M. and Hlaváč, V. *Ten lectures on statistical and structural pattern recognition*. Kluwer Academic Publishers, 2002.

Tortorella, F. An optimal reject rule for binary classifiers. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2000.

Vapnik, V. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.

Wang, W., Wang, A., Tamar, A., Chen, X., and Abbeel, P. Safer classification by synthesis. *CoRR*, abs/1711.08534, 2017. URL http://arxiv.org/abs/1711.08534.

Wu, T., Lin, C., and Weng, R. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.