Active learning algorithm through the lens of rejection arguments

Christophe Denis, Mohamed Hebiri, Boris Ndjia Njike, Xavier Siebert

Active learning algorithm through the lens of rejection arguments.

Christophe Denis[(1)], Mohamed Hebiri[(1)], Boris Ndjia Njike[(2)], Xavier Siebert[(2)]

[(1)] LAMA, Université Gustave Eiffel, France

[(2)] Mathematics and Operational Research University of Mons, Belgium

August 30, 2022

Abstract

Active learning is a paradigm of machine learning which aims at reducing the amount of labeled data needed to train a classifier. Its overall principle is to sequentially select the most informative data points, which amounts to determining the uncertainty of regions of the input space. The main challenge lies in building a procedure that is computationally efficient and that offers appealing theoretical properties; most of the current methods satisfy only one or the other. In this paper, we use the classification with rejection in a novel way to estimate the uncertain regions. We provide an active learning algorithm and prove its theoretical benefits under classical assumptions. In addition to the theoretical results, numerical experiments have been carried out on synthetic and non-synthetic datasets. These experiments provide empirical evidence that the use of rejection arguments in our active learning algorithm is beneficial and allows good performance in various statistical situations.

Keywords: active learning, rejection, nonparametric learning, classification

1 Introduction

The aim of machine learning consists in designing learning models that accurately maps a set of inputs from a space $\mathcal{X}$ called *instance space* to a set of outputs $\mathcal{Y}$ called *label space*. Nowadays, with the data deluge, obtaining a powerful learning model requires a lot of data from $\mathcal{X}$ to be labeled, which is time consuming in many modern applications such as speech recognition or text classification. This motivated the development of other paradigms beyond classical prediction tasks. In this paper, we focus on prediction in the binary classification setting, that is $\mathcal{Y} = \{0, 1\}$. In this framework, one of the most studied techniques to deal with this specificity is the iterative supervised learning procedure called *active learning* (Cohn *et al.*, 1994; Castro & Nowak, 2008; Balcan *et al.*, 2009; Hanneke, 2011; Locatelli *et al.*, 2017, 2018) that aims at reducing the data labeling effort by carefully selecting which data need to be labeled. The goal of *active learning* is to achieve a high rate of correct predictions while using as few labeled data as possible. One of the key principles of active learning is to identify at each step the region of the instance space where the label requests should be made, called *uncertain region* in this paper, also known as *disagreement region* in the active learning literature (Hanneke, 2007; Balcan *et al.*, 2009; Dasgupta, 2011). Many techniques have been developed to this aim, both in parametric (Cohn *et al.*, 1994; Hanneke, 2007; Balcan *et al.*, 2009; Beygelzimer *et al.*, 2009; Hanneke *et al.*, 2014) and nonparametric setting (Minsker, 2012; Locatelli *et al.*, 2017, 2018).

In this paper, we are particularly interested in the nonparametric setting, where several computational difficulties have so far hampered the practical implementation of the proposed algorithms. For example, (Minsker, 2012) provides interesting theoretical results which partly motivated Locatelli *et al.* (2017, 2018) as well as the present work, but it fails to provide a computationally efficient way to estimate the uncertain region.

To overcome these shortcomings, we present a new active learning algorithm using the paradigm called *rejection*. The latter typically allows the learning models to evaluate their confidence in each prediction and to possibly abstain from labeling an instance (*i.e.*, "reject" this instance) when the confidence in the prediction of its label is too weak. This rejection will however be used in a novel way in this work to conveniently compute the uncertain region, as explained below.

Rejection and active learning typically differ on how they are interested in this uncertain region. In rejection, the interest in the uncertain region appears *after* the design of a learning model, that rejects a test point in order to avoid a misprediction. This is very useful in some applications such as medical diagnosis where a misprediction can be dramatic. However, in active learning, the uncertain region is used *during* the training process to progressively improve the model's performance by requesting labels where the classification is difficult.

In our algorithm, we use rejection at each step k of the training process to estimate the uncertain region $A_k \subset \mathcal{X}$ based on the information gathered up to this step. Then some points are sampled from the region A_k and their labels are requested. Based on these labeled examples, an estimator $\hat{f}_k$ is provided, that is then used to assess for each $x \in A_k$ the confidence in the prediction. The points where the confidence is low are rejected and are considered to form the next uncertain region A_{k+1}, thereby progressively reducing the part of instance space $\mathcal{X}$ on which a model remains to be constructed. We study the rate of convergence with respect to the excess-risk of our nonparametric active learning algorithm based on histograms under classical smoothness assumptions. It turns out that combining active learning sampling together with rejection allows for optimal rates of convergence. Using numerical experiments on several datasets we also show that our active learning process can be efficiently applied to any off-the-shelf machine learning algorithm.

The paper is organized as follows : in Section 2 we provide the background notions of active learning and rejection separately, then review some recent works that proposed to combine these two notions, although in a way that differs from ours. Then we describe our algorithm in Section 3 along with the theoretical guarantees about its rate of convergence. Practical considerations to take into account when applying our algorithm are discussed in Section 4. Numerical experiments are presented in Section 5 and we conclude the paper along with some perspectives for future work in Section 6. The full proof of our theoretical result is relegated to the Appendix.

2 Background

In this Section we review the literature related to active learning in Section 2.1, and the reject option framework 2.2. Thereafter, in Section 2.3 we provide a review on the use of the rejection in the context of active learning.

2.1 Active learning

Given an i.i.d. sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from an unknown probability distribution P defined on $\mathcal{X} \times \mathcal{Y}$, the classification problem consists in designing a map $g : \mathcal{X} \longrightarrow \mathcal{Y}$ from the instance space to the label space. However, building such mapping might become a tricky task in particular situations where the labeling process of input instances are only available through time-consuming or expensive requests to a so-called oracle. In such applications, one might however have access to a huge amount

of unlabeled data from the instance space. This motivated the use of the *active learning* paradigm (Cohn *et al.*, 1994) that aims at reducing the data labeling effort by carefully selecting which data to label.
Active learning algorithms were initially designed according to somewhat heuristic principles (Settles, 1994) without theoretical guarantees on the convergence nor on the expected gain with respect to classical "passive" learning. The theory of active learning has then gradually developed (Cohn *et al.*, 1994; Freund *et al.*, 1997; Balcan *et al.*, 2009; Hanneke, 2007; Dasgupta *et al.*, 2007; Castro & Nowak, 2008; Minsker, 2012; Hanneke & Yang, 2015; Locatelli *et al.*, 2018, 2017; Kpotufe *et al.*, 2022).
We are particularly interested in the nonparametric setting, where regularity and noise assumptions are made on the regression function. Two types of regularity assumptions are made on the regression function. The first one was introduced in the seminal work by (Castro & Nowak, 2008) and was also used in (Locatelli *et al.*, 2018), where it is assumed that the decision boundary $\{x,\ \eta(x) = \frac{1}{2}\}$ (where η is the regression function) is the graph of a smooth function. The second one, which was used in (Minsker, 2012; Locatelli *et al.*, 2017), assumes that the whole regression function is smooth. In this work, we will use similar regularity assumption as in (Minsker, 2012). Besides, the noise margin assumption corresponds to the so-called *Tsybakov noise condition*, and it was observed that it corresponds to the situation in which active learning can outperforms passive learning (Castro & Nowak, 2008).
In this work, we design an efficient active learning algorithm, similar to that considered in (Minsker, 2012), but handling the uncertain region in an explicit and computationally tractable way using rejection.

2.2 Classification with reject option

In the present contribution, we borrow some techniques from learning with reject option. Indeed, as detailed in Section 3, a core component of our active strategy relies on the confidence we have on labels of the input instances. In contrast to the classical statistical learning framework where a label is provided for each observation $x \in \mathcal{X}$, learning with reject option is based on the idea that an observation for which the confidence on the label is not high enough should not be labeled. From this perspective, given a prediction function $g : \mathcal{X} \to \mathcal{Y}$, an instance $x \in \mathcal{X}$ can be either classified and the corresponding label is $g(x)$ or rejected and no label is provided for x (according to the literature, the output for x is $\emptyset$ or any symbol as $\oplus$ meaning reject). A classifier with reject option $\tilde{g}$ is then a measurable mapping $\tilde{g} : \mathcal{X} \to \mathcal{Y} \cup \{\oplus\}$. Reject option has been first introduced in the classification setting in (Chow, 1957). More recently, and since the development of *conformal prediction* in (Vovk *et al.*, 1999, 2005), reject option has become more popular and has been brought up to date to meet the current challenges. The paper by (Herbei & Wegkamp, 2006) proposed the first statistical analysis of a classifier based on reject option. After these pioneer works, more papers on reject options appeared (e.g., (Naadeem *et al.*, 2010; Grandvalet *et al.*, 2009; Yuan & Wegkamp, 2010; Lei, 2014; Cortes *et al.*, 2016; Denis & Hebiri, 2019) and references therein). They mainly differ on the way they take into account the reject option. In particular, we can distinguish three main approaches: i) use the reject option to unsure a predefined level of coverage; ii) use the reject option to unsure a pre-specified proportion of rejected data; iii) consider a loss that balances the coverage and the proportion of rejected data. It has been established that, while there is no best strategy, controlling the coverage requests more labeled data than controlling the rejection rate, which in turn asks more (unlabeled) data that the last strategy that does the trade-off. On the other hand this last approach does not control any of the two parameters.
Reject option has also been used in different contexts, such as in regression (Vovk *et al.*, 2005; Denis *et al.*, 2020) or algorithmic fairness (Schreuder & Chzhen, 2021). These papers show how reject option can be used to efficiently solve issues that are intrinsic to the problem.

2.3 Active learning with reject option

Most active learning schemes mentioned in Section 2.1 attempt to find the most "informative" samples in a region close the decision boundary, called *uncertain region* or *disagreement region*. Some recent works have refined this idea by adding an option to abstain from labeling the points (*i.e.*, reject) that are considered too close to the decision boundary.

Although the intersection of rejection and active learning seems natural, their combination is fairly recent. Current studies can be grouped into two differents settings: the first one is focused on using reject option for improving performance guarantees of some standard active learning algorithms (Puchkin & Zhivotovskiy, 2021; Zhu & Nowak, 2022) and the second one is focused on providing a classifier which takes into account reject option (Shekhar *et al.*, 2021; Shah & Manwani, 2020), similarly to the standard reject option setting (Herbei & Wegkamp, 2006; Denis & Hebiri, 2019).

In the first setting, (Puchkin & Zhivotovskiy, 2021) considered the parametric framework, particularly the model misspecification. That is, given a class of classifiers $\mathcal{F}$ (which possibly do not contain the Bayes classifier), the aim is to find an estimator $\hat{f}$ which achieves minimum excess error of classification. By using the reject option, (Puchkin & Zhivotovskiy, 2021) proved that exponential savings in the number of label requests are possible in model misspecification under Massart noise assumption (Massart & Nédélec, 2006). Their algorithm is related to the disagreement-based approach (Hanneke, 2007; Balcan *et al.*, 2009) and outputs an improper classifier $\hat{f}$, that is $\hat{f} \notin \mathcal{F}$ possibly. The work of (Puchkin & Zhivotovskiy, 2021) was extended by (Zhu & Nowak, 2022) which provides a more efficient active learning algorithm that overcomes the difficulty of computing the uncertain region. In (Zhu & Nowak, 2022), the authors build a classifier based on the rejection rule with exponential saving in labels, for which they establish risk bounds in a general parametric setting. At each trial, the classifier does not label points for which the doubt is substantial. This decision of abstaining from classifying a point is taken by considering a set of "good" classifiers among a parametric class of functions. In particular, a point is rejected if all "good" classifiers consider it as a difficult point, that is, the corresponding score is within the interval $[1/2-\gamma, 1/2+\gamma]$, where γ is a (small) positive real value. However an analysis of this algorithm sheds light on three arguments. First, the score at point x should be evaluated for all "good" functions in the class. Second, tuning the parameter γ is not discussed and it might be tricky. Finally, the empirical performance of the proposed algorithm is not considered in the paper.

In the second setting, (Shekhar *et al.*, 2021), considered the nonparametric framework under some smoothness and margin noise assumptions. The authors designed an active learning algorithm which outputs a classifier that takes into account the reject option in a standard way as in (Denis & Hebiri, 2019) by deciding not to label the instances which are located near to the decision boundary. In particular, the final outputted algorithm is a classifier with reject option. In their framework, they derived rates of convergence for an excess-risk dedicated to the reject option framework and showed that these rates are better to those obtained by the passive learning counterpart (Denis & Hebiri, 2019). However it is not obvious in this setting to obtain computationally tractable algorithms, among others because the hypothesis class needs to be restricted. In contrast, in the present paper, we focus on the classical active problem and derive rates of convergence for this problem, along with a practical implementation of the algorithm.

2.4 Contributions

The recent works mentioned in Section 2.3 (Puchkin & Zhivotovskiy, 2021; Shekhar *et al.*, 2021; Zhu & Nowak, 2022) provide interesting theoretical contributions showing the interest of combining active learning and reject option. However the practical implementation of the related algorithms is not straightforward, notably because it is computationally difficult to estimate the uncertain region.

In this work, we use a peculiar combination of the rejection and active learning to propose an active learning which is easy to compute in practice. More precisely, our contributions are threefold:

- We transform the typical classification with reject option framework (from Sections 2.2 and 2.3) to estimate the so-called uncertain region in a novel way. Not only does this methodology provide a computationally efficient algorithm for active learning, but it also can be remarkably applied to any off-the-shelf machine learning algorithm. This is a twofold major improvement over (Minsker, 2012).
- Beyond the appealing numerical properties of our procedure, we show that it achieves optimal rates of convergence for the misclassification risk and the active sampling under classical assumptions in this setting.
- We illustrate the benefit of our method in synthetic and real datasets.

3 Active learning algorithm with rejection

In this section, after introducing some general notations and definitions, we present our algorithm in a somewhat informal way, and then provide the theoretical guarantees under some classical assumptions.

3.1 Notations and definitions

Throughout this paper $\mathcal{X}$ denotes the instance space and $\mathcal{Y} = \{0, 1\}$ is the label space. Let P be the joint distribution of (X, Y). We denote by Π the marginal probability over the instance space and by $\eta(x) = P(Y = 1|X = x)$ the regression function. The performance of a classification rule $g : \mathcal{X} \mapsto \{0, 1\}$ is measured through the misclassification risk $R(g) = P\left(g(X) \neq Y\right)$. With this notation, the Bayes optimal rules that minimises the risk R over all measurable classification rules (Lugosi, 2002) is given by $g^*(x) = \mathbb{1}_{\{\eta(x) \geq 1/2\}}$ and we have:

$$R(g^*) = 1 - \mathbb{E}_\Pi(f^*(X)) \ ,$$

where $f^*(\cdot) = \max(\eta(\cdot), 1 - \eta(\cdot))$ is called *score function*. For any classification rule g, the excess risk is given by

$$R(g) - R(g^*) = 2\mathbb{E}\left[\left|\eta(X) - \frac{1}{2}\right| \mathbb{1}_{\{g(X) \neq g^*(X)\}}\right] \ . \tag{3.1}$$

In this work, we consider the following active sampling scheme. For each $A \subset \mathcal{X}$, and $M \geq 1$, we can sample $(X_i, Y_i)_{1 \leq i \leq M}$ i.i.d. random variables such that

1. for all $i = 1, \ldots, M$, X_i is distributed according to $\Pi(.|A)$;
2. conditional on X_i, the random variable Y_i is distributed according to a Bernoulli random variable with parameter $\eta(X_i)$.

As is commonly done in the active learning setting, we assume that the marginal distribution of X is known (Minsker, 2012; Locatelli *et al.*, 2017). In the next paragraph, we describe our active algorithm for classification. As important tools that nicely merge the active sampling and the use of the rejection, we will pay a particular attention to the definition of the uncertain region and the rejection rate.

3.2 Overall description of the algorithm

With a fixed number of label requests N (called the budget), our overall objective is to provide an active learning algorithm which outputs a classifier that performs better than its passive counterpart. The framework that we consider (Algorithm 1) is inspired from that developed in (Minsker, 2012), in which we incorporate rejection to estimate the uncertain region.
In the following, let $(\varepsilon_k)_{k\geq 0}$ be a sequence of positive numbers. Let $(N_k)_{k\geq 0}$ be a sequence defined such that $N_0 = \sqrt{N}$ and $N_{k+1} = \lfloor c_N N_k \rfloor$ with $c_N > 1$ (e.g., $c_N = 1.2$ in Section 5). Furthermore, we consider $A_0 = \mathcal{X} = [0,1]^d$ the initial uncertain region, and thus $\varepsilon_0 = 1$. We construct a sequence of uncertain regions $(A_k)_{k\geq 1}$ and for $k \geq 1$, an estimator $\hat{\eta}_k$ of η on A_k is provided.
First, our algorithm performs an initialization phase:

- Initially, the learner requests the labels Y of N_0 points $X_1, \ldots, X_{N_0}$ sampled in A_0 according to $\Pi_0 = \Pi$.
- Based on the initial labeled data $\mathcal{D}_{N_0} = \{(X_1, Y_1), \ldots, (X_{N_0}, Y_{N_0})\}$, an estimator $\hat{\eta}_0$ of η on A_0 is computed and an initial classifier $g_{\hat{\eta}_0} = \mathbb{1}_{\{\hat{\eta}_0 \geq 1/2\}}$ is provided.
- An estimator of the score function $\hat{f}_0(x) = \max(\hat{\eta}_0(x), 1 - \hat{\eta}_0(x))$ associated to $\hat{\eta}_0$ is computed.

Afterwards, our algorithm iterates over a finite number of steps until the label budget N has been reached. Step $k \geq 1$ is described below.

- Based on the previous uncertain region A_{k-1}, a constant λ_k is computed such that conditional on the data
$$\lambda_k = \max\left\{t,\ \ \Pi\left(\hat{f}_{k-1}(X) \leq t | A_{k-1}\right) \leq \varepsilon_k\right\} \ , \tag{3.2}$$
These $(\varepsilon_k)_{k\geq 0}$ define explicitly the *sequence of the rejection rates* (Denis & Hebiri, 2019).
- This constant λ_k is used to construct the *current uncertain region* A_k which is the set where the previous classifier $g_{\hat{\eta}_{k-1}}(\cdot) = \mathbb{1}_{\{\hat{\eta}_{k-1}(\cdot) \geq 1/2\}}$ might fail and thus abstains from labeling :
$$A_k = \{x \in A_{k-1},\ \ \hat{f}_{k-1}(x) \leq \lambda_k\} \ ,$$
where $\hat{f}_{k-1}(x) = \max(\hat{\eta}_{k-1}(x), 1 - \hat{\eta}_{k-1}(x))$.
- According to $\pi(.|A_k)$ the learner samples i.i.d. $(X_i, Y_i), i = 1, \ldots, \lfloor N_k \varepsilon_k \rfloor$ used to compute an estimator $\hat{\eta}_k$ of η on A_k.
- The learner updates the classifier over the whole space $\mathcal{X}$ as follows
$$\hat{\eta} = \sum_{j=0}^{k-1} \hat{\eta}_j \mathbb{1}_{\{A_j \setminus A_{j+1}\}} + \hat{\eta}_k \mathbb{1}_{\{A_k\}} \ .$$

After the iteration process, the resulting active classifier with rejection is defined point-wise as
$$\hat{g}(x) = \mathbb{1}_{\{\hat{\eta}(x) \geq 1/2\}} \ . \tag{3.3}$$

3.3 Theoretical guarantees

This section is devoted to the theoretical properties of the proposed procedure under common assumptions which are presented in Section 3.3.1. Thereafter, we state our main result in Section 3.3.2 that mainly shows that our algorithm achieves an optimal rate of convergence for the excess-risk when the considered classifier is the histogram rule.

3.3.1 Assumptions

We assume that $\mathcal{X} = [0,1]^d$ and consider two assumptions that are widely considered for the study of rates convergence in the passive (Audibert & Tsybakov, 2007; Gadat *et al.*, 2016) or active settings (Minsker, 2012; Locatelli *et al.*, 2017).

Assumption 3.1 (Smoothness assumption)**.** *The regression function η is s-Lipschitz-continuous for some $s \geq 0$, that is, for all $x, z \in [0,1]^d$:*

$$|\eta(x) - \eta(z)| \leq s. \parallel x - z \parallel_\infty \quad .$$

Assumption 3.2 (Strong density assumption)**.** *The marginal probability admits a density p_X and there exist constants $\mu_{min}, \mu_{max} > 0$ such that for all $x \in [0,1]^d$ with $p_X(x) > 0$, we have:*

$$\mu_{min} \leq p_X(x) \leq \mu_{max} \quad .$$

Assumption 3.1 imposes the regularity of the regression function η while Assumption 3.2 ensures in particular that the marginal distribution of X admits a density which is bounded from below. Furthermore, we also assume that $f(X)$ admits a bounded density.

Assumption 3.3 (Score regularity assumption)**.** *Let $f(x) = \max(\eta(x), 1-\eta(x))$ be the score function. The random variable $f(X)$ admits a bounded density (bounded by $C > 0$).*

Assumption 3.3 has two important consequences. The first one is that the cumulative distribution function F_f of $f(X)$ is Lipschitz. The second one is that the so-called Margin assumption (Tsybakov, 2004) is fulfilled with margin parameter $\alpha = 1$. This Margin assumption is also considered in (Minsker, 2012) for the study of optimal rates of convergence in the active learning framework.

3.3.2 Rates of convergence

In this section, we present our main theoretical result (Theorem 3.5) which highlights the performance of our algorithm. While our methodology can handle any machine learning algorithm for the estimation of the regression function η, we provide theoretical guarantee with the histogram rule (whose definition is recalled in Definition 3.4) for the estimation of the regression function at each step of the procedure described in Section 3.2, as in (Minsker, 2012). For completeness, we provide the full proof of our result in this particular case in the Appendix.
Let us denote by $\mathcal{C}_r = \{R_i,\ i = 1, \ldots, r^{-d}\}$ a cubic partition of $[0,1]^d$ with edge length $r > 0$.

Definition 3.4 (Histogram rule)**.** *Let A be a subset of $[0,1]^d$. Consider a labeled sample $\mathcal{D}_{N_A} = \left\{(X_1^A, Y_1), \ldots, (X_{N_A}^A, Y_{N_A})\right\}$ of size $N_A \geq 1$, such that X_i^A $(i = 1, \ldots, N_A)$ is distributed according to $\Pi(.|A)$. The histogram rule on A is defined as follows. Let $R_i \in \mathcal{C}_r$ with $R_i \cap A \neq \emptyset$. For all $x \in R_i$,*

$$\hat{\eta}_{A,N_A,r}(x) = \frac{\Pi(A)}{\Pi(R_i)} \frac{1}{N_A} \sum_{j=1}^{N_A} Y_j \mathbb{1}_{\{X_j \in R_i\}} \quad .$$

It is known that in the passive framework, the histogram rule achieves optimal rates of convergence (Devroye *et al.*, 1996).

Theorem 3.5. *Let N be the label budget, and $\delta \in \left(0, \frac{1}{2}\right)$. Let us assume that Assumptions 3.1, 3.2, and 3.3 are fulfilled. At each step $k \geq 0$ of the algorithm presented in Section 3.2, we consider*

i) $\hat{\eta}_k := \hat{\eta}_{A_k, \lfloor N_k \Pi(A_k) \rfloor, r_k}$, *with* $r_k = N_k^{-1/(d+2)}$,

(ii) and define $(\varepsilon_k)_{k\geq 0}$ *as* $\varepsilon_0 = 1$*, and for* $k \geq 1$*,* $\varepsilon_k = \min\left(1, \log\left(\frac{N}{\delta}\right)\log(N)N_{k-1}^{-1/(2+d)}\right)$.

Then with probability at least $1-\delta$*, the resulting classifier defined in Equation*(3.3) *satisfies*

$$R(g_{\hat{\eta}}) - R(g^*) \leq \widetilde{O}\left(N^{-\frac{2}{1+d}}\right), \tag{3.4}$$

where $\widetilde{O}$ *hides some constants and logarithmic factors.*

The above result calls for several comments. First, our active classifier $\hat{g}$ based on the histogram rule is optimal for the active sampling *w.r.t.* the misclassification risk up to some logarithmic factors (see (Minsker, 2012) for the minimax rates, by considering Lipschitz regression function and the margin parameter equal to 1. This rate is better than the classical minimax rate in passive learning under the strong density assumption which is of order $N^{-\frac{2}{2+d}}$, see for instance Audibert & Tsybakov (2007). Second, the sequence of the rejection rates $(\varepsilon_k)_{k\geq 0}$ should be chosen in an optimal manner guided by our theoretical findings. In particular, for each k, the value of ε_k is of the same order as an upper bound on the error *w.r.t.* the ℓ_∞-norm of $\hat{\eta}_{k-1}$, valid with high probability. This value of the ε_k is also linked to the probability of the uncertain region in the procedure proposed by Minsker (2012). However, the major different with the latter reference is that our rejection rate is explicit and then our algorithm can be efficiently computed due to the use of rejection arguments to determine the uncertain regions. Finally, let us notify that our work can easily be extended for Hölder regression functions with parameter β. Indeed, for $\beta \geq 1$, we can consider a similar estimator as that introduced in Definition 3.4 with higher order histogram rule using smoothing kernel (Giné & Nickl, 2021).

Remark 3.6. *Theorem 3.5 is established assuming the knowledge of the marginal distribution of* X*. This is a classical assumption in active learning that helps for sampling. However, it is possible to extend our result to unknown distributions at the price of an additional unlabeled sample and then an additional factor* $1/\sqrt{\text{size of the unlabeled sample}}$*.*

In view of the above remark, we discuss the practical implementation of our proposed algorithm in the following section.

4 Practical considerations

Some practical aspects of the procedure are discussed in Section 4.1 and a simple numerical illustration is provided in Section 4.2. The full numerical experiments are presented in Section 5.

4.1 Uncertain region

In this section, we discuss the effective computation of the uncertain regions. Let $k \geq 1$ represent the current step k of our algorithm. We denote by $\mathcal{D}_M = \{X_1, Y_1), \ldots, (X_M, Y_M)\}$ the data that have been sampled until step k. The random variable $\hat{f}_{k-1}$ is the score function built at step $k-1$.
The construction of the uncertain region A_k relies on λ_k which is solution of Equation (3.2). First of all, we randomize the score function $\hat{f}_{k-1}$ by introducing a variable ζ distributed according to a Uniform distribution on $[0, u]$ independent of $\mathcal{D}_M$ and by defining the randomized score function $\tilde{f}_{k-1}$ as

$$\tilde{f}_{k-1}(X, \zeta) = \hat{f}_{k-1}(X) + \zeta \ .$$

Considering the randomized score $\tilde{f}_{k-1}$ instead of $\hat{f}_{k-1}$ ensures that conditionally on $\mathcal{D}_M$, the cumulative distribution function of $\tilde{f}_{k-1}(X, \zeta)$, denoted by $F_{\tilde{f}_{k-1}}$, is continuous. Therefore, it implies that

$$\tilde{\lambda}_k = \max\left\{t, \ \Pi\left(\tilde{f}_{k-1}(X) \leq t | A_{k-1}\right) \leq \varepsilon_k\right\} = F^{-1}_{\tilde{f}_{k-1}}(\varepsilon_k) \ .$$

Hence, $\tilde{\lambda}_k$ is expressed simply as the ε_k-quantile of the c.d.f. $F_{\tilde{f}_{k-1}}$. To preserve the statistical properties of $\hat{f}_{k-1}$, the parameter u is chosen sufficiently small (*e.g.,* $u \to 0$).
Note that the computation of the c.d.f. $F_{\tilde{f}_{k-1}}$ requires the knowledge of the marginal distribution of X. In practice, this distribution may be unknown. In a second step, based on a *unlabeled* dataset $\mathcal{D}^U_{M_k} = \{X_i, i = 1, \ldots, M_k\}$ with $X_i \sim \Pi(.|\hat{A}_{k-1})$, and $(\zeta_1, \ldots, \zeta_{M_k})$ i.i.d. copies of ζ, we consider an estimator $\hat{\lambda}_k$ of $\tilde{\lambda}_k$ defined as follows

$$\hat{\lambda}_k = \hat{F}^{-1}_{\tilde{f}_{k-1}}(\varepsilon_k),$$

where conditionally on the data, $\hat{F}_{\tilde{f}_{k-1}}$ is the empirical c.d.f. of the random variable $\tilde{f}_{k-1}(X, \zeta)$:

$$\hat{F}_{\tilde{f}_k}(t) = \frac{1}{M_k} \sum_{i=1}^{M_k} \mathbb{1}_{\{\tilde{f}_k(X_i,\zeta_i) \leq t\}} \ .$$

Furthermore, the unlabeled set $\mathcal{D}^U_{M_k}$ is assumed to be independent of $\mathcal{D}_M$, and since it remains unlabeled, it does not contribute to the budget.
Formally, the uncertain region A_k is then defined as follows

$$A_k = \left\{(x, \zeta) \in \mathcal{X} \times [0, u], \ \ \tilde{f}_{k-1}(x, \zeta) \leq \hat{\lambda}_k \right\} \ .$$

Therefore, $X_{M+1} \sim \Pi\,(.|A_k)$, is sampled from Π such that $\tilde{f}_{k-1}(X_{M+1}, \zeta) \leq \hat{\lambda}_k$ with ζ distributed according to $\mathcal{U}_{[0,u]}$.

4.2 Illustrative example

For illustrative purposes, a two-dimensional dataset of 10^6 data points was generated using a regression function $\eta(x_1, x_2) = \frac{1}{2}(1 + \sin(\frac{\pi x_2}{2}))$. We chose the estimators $\hat{\eta_k}$ to be linear, to make the comparison with the best linear classifier ($x_2 = 0$) straightforward. The budget was set to $N = 5000$, and the sequences of N_k and ε_k were chosen as $N_k = \lfloor 1.2\, N_{k-1} \rfloor$ and $\varepsilon_k = 0.95\, \varepsilon_{k-1}$, starting with $N_0 = \lfloor \sqrt{N} \rfloor$ and $\varepsilon_0 = 1$. The parameter M_k was set to 150. A discussion of this choice of parameters can be found in Section 5.1.
Figure 1 represents the situation after the step $k = 2$ of the algorithm. At step $k = 1$ and $k = 2$, λ_k has been computed using (3.2), which allows to classify the points in $\hat{A}_{k-1} \setminus \hat{A}_k$ (represented in black for $k = 1$ and in brown for $k = 2$). For visualization purposes, the points remaining in $\hat{A}_2$ have been colored according to their labels ($y = 1$ in green and $y = 0$ in blue), even though these labels are unknown at this step of the algorithm. The yellow points are those in $\hat{A}_2$ whose label has already been requested to the oracle. At subsequent steps, points in A_k are selected according to the rejection rates shown in the center part of Figure 1, which shows the theoretical reject rates (ε_k, defined in Algorithm 1) in blue and the experimental ones ($\hat{\varepsilon}_k$, counted as the number of points effectively rejected) in red. The latter were computed by repeating the simulations 10 times, to present the average results along with the standard deviations in grey. As a whole, the rejection rate is well estimated with only $M_k = 150$ unlabeled samples. However, the standard deviations indicates that the rejection rate is harder to control towards the end of the algorithm, because less points are available to estimate ε_k.
The resulting learning curves for passive and active procedures are represented on the right of Figure 1. As expected with this simplistic illustrative dataset, using active learning does not provide a substantial advantage in the long run (test precision $= 0.817 \pm 0.005$ for active; 0.816 ± 0.003 for passive), because the optimal classifier is relatively easy to find in passive learning, even with noisy data. However, the right panel of Figure 1 shows that for a given small budget (e.g., $N < 500$), active learning converges faster than passive learning. This will be further examined in Section 5.

Algorithm 1: Active learning with rejection

Input: label budget N

Initialization

The uncertain region $\hat{A}_0 = [0,1]^d$

$N_0 = \lfloor \sqrt{N} \rfloor$

$k = 1$

$B = N_0$

$\varepsilon_0 = 1$, for all $k \geq 1$, define the rejection rate ε_k

for $i = 1$ ***to*** N_0 **do**

  Sample i.i.d $(X_{i,0}, Y_{i,0})$ with $X_{i,0} \sim \Pi$

$\mathcal{D}_{N_0} = \{(X_{1,0}, Y_{1,0}) \ldots, (X_{N_0,0}, Y_{N_0,0})\}$

Based on $\mathcal{D}_{N_0}$, compute an estimator $\hat{\eta}_{\mathcal{D}_{N_0}}$.

$\hat{\eta}_0 := \hat{\eta}_{\mathcal{D}_{N_0}}$

while $B + \lfloor N_k \varepsilon_k \rfloor \leq N$ **do**

  Sample i.i.d $D^U_{M_k} = \{X_i, i = 1, \ldots, M_k\}$ with $X_i \sim \Pi(.|\hat{A}_{k-1})$.

  Based on $D^U_{M_k}$, compute $\hat{\lambda}_k$ such that $\widehat{\mathbb{P}}(\hat{f}_{k-1} \leq \hat{\lambda}_k | \hat{A}_{k-1}) = \varepsilon_k$

  $\hat{A}_k := \{x \in \hat{A}_{k-1}, \ \ \hat{f}_{k-1}(x) \leq \hat{\lambda}_k\}$

  $N_k = c_N N_{k-1}$

  for $i = 1$ ***to*** $\lfloor N_k \varepsilon_k \rfloor$ **do**

    Sample i.i.d $(X_{i,k}, Y_{i,k})$ with $X_{i,k} \sim \Pi(.|\hat{A}_k)$

  $\mathcal{D}_{N_k} = \{(X_{1,k}, Y_{1,k}) \ldots, (X_{\lfloor N_k \varepsilon_k \rfloor, k}, Y_{\lfloor N_k \varepsilon_k \rfloor, k})\}$

  Based on $\mathcal{D}_{N_0}$, compute an estimator $\hat{\eta}_{\mathcal{D}_{N_k}}$

  $\hat{\eta}_k := \hat{\eta}_{\mathcal{D}_{N_k}}$

  $\hat{\eta} = \sum_{j=0}^{k-1} \hat{\eta}_j \mathbb{1}_{\{\hat{A}_j \setminus \hat{A}_{j+1}\}} + \hat{\eta}_k \mathbb{1}_{\{\hat{A}_k\}}$

  $B = B + \lfloor N_k \varepsilon_k \rfloor$

  $k = k + 1$

Output: $\hat{g}_{\hat{\eta}}(x) = \mathbb{1}_{\{\hat{\eta}(x) \geq 1/2\}}$ for all $x \in [0,1]^d$

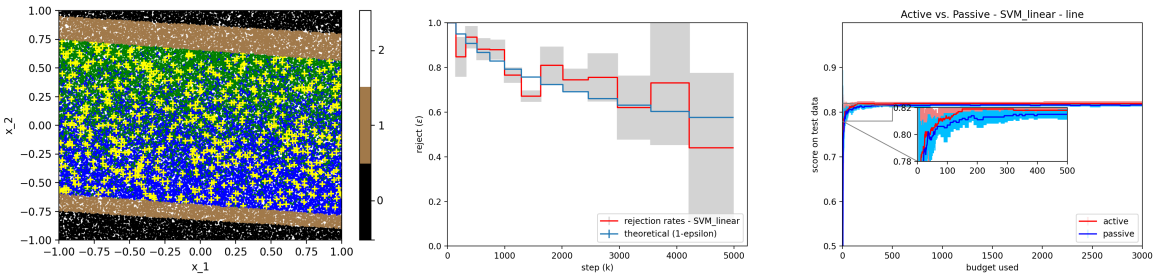

Figure 1: Left: Illustrative dataset after the step $k = 2$ of the algorithm. The points in black belong to $\hat{A}_0 \setminus \hat{A}_1$ and the brown ones to $\hat{A}_1 \setminus \hat{A}_2$. In $\hat{A}_2$ are the yellow points whose label have been requested to the oracle and the remaining points in green and blue correspond to $y = 1$ and $y = 0$, respectively. Center: theoretical (ε_k, blue) and experimental ($\hat{\varepsilon}_k$, red with error bars in grey) rejection rates. Right: active vs. passive learning curves.

5 Numerical experiments

5.1 Parameters choice and sampling strategy

This Section discusses some aspects of the practical implementation of our algorithm.

Parameters choice To perform numerical experiments, a few parameters of our model have to be set. First, the sequence of rejection rates was defined such that $\varepsilon_{k+1} = c_\varepsilon \varepsilon_k$, with $\varepsilon_0 = 1$ and $c_\varepsilon \in]0, 1[$. If c_ε is small, the uncertain region $\hat{A}_k$ will be small, which corresponds to an "aggressive" strategy where many points are considered to be correctly classified at each step. Conversely, if c_ε is large, the strategy will be more "conservative". Second, the constant c_N defines the sequence N_k as $N_k = \lfloor c_N N_{k-1} \rfloor$ and thus the number of points asked to the oracle at step k ($\lfloor N_k \hat{\varepsilon}_k \rfloor$ on line 17 of Algorithm 1). If c_N is large, the algorithm will use many points at each step, thereby consuming the budget faster. A larger budget therefore allows a larger c_N. Third, the number of points to build the initial classifier is theoretically set to $N_0 = \lfloor \sqrt{N} \rfloor$. In practice, this number can be increased to get a better estimate of $\hat{\eta}_0$. Using a larger N_0 will however consume the budget faster. Third, M_k unlabeled data points in $D^U_{M_k}$ are used at each step to estimate $\hat{\lambda}_k$. If M_k is large, the estimation of $\hat{\lambda}_k$ will be more accurate. As these M_k points remain unlabeled, they do not contribute to the budget, and M_k could in principle be large. The only restriction is that at each step k these (unlabeled) points have to be sampled independently of the (labeled) points asked to the oracle, it indirectly limits the number of points available to the oracle. Several experiments (results not shown) indicate that $M_k \geq 100$ provides a reasonable estimate of $\hat{\lambda}_k$. Finally, the parameter u in Section 4.1 has been set to 10^{-5}. Its precise value does not affect much the results, as long as it remains close to 0.
Unless otherwise stated, our numerical experiments were performed using a "conservative approach, with the parameters discussed above set to $c_\varepsilon = 0.95$, $c_N = 1.2$, $N_0 = 2\lfloor \sqrt{N} \rfloor$ and $M_k = 150$.

Sampling strategy We designed a sampling strategy that re-uses points whenever possible, using two recycling procedures explained below. This is not so important in our numerical experiments with synthetic data (Section 5.2), where 10^5 data points are used to mimic the theoretical situation with an "infinite" pool of data. However it can become crucial in practical applications with limited labeled data, as in the non-synthetic datasets used in Section 5.3.
The first recycling procedure is that the unlabeled points from step $k-1$ will be re-used at step k. This does not invalidate our theory just because of the additive form of the risk over cells A_k. Indeed, our trained estimator has the form $\hat{g}(\cdot) = \sum_k \hat{g}_k(\cdot) \mathbb{1}_{A_k}(\cdot)$ and then its overall risk $R(\hat{g})$ can be decomposed on the different regions A_k (by conditioning on the data used to approximate the region from the previous iteration).
The second recycling procedure is that the data already labeled by the oracle at previous iterations (up to $k-1$ included) are reused to train $\hat{\eta}_k$, as long as they belong to the region A_k. A similar procedure was used in (Urner *et al.*, 2013). This allows to improve the estimation of $\hat{\eta}_k$ and to limit the budget consumption. This sampling strategy is permitted because of the expression of the estimator and the decomposition of the risk as noted above. It is particularly useful in practical applications where the total amount of labeled data is limited.

5.2 Synthetic datasets

Setting These numerical experiments were performed using 10^5 data points with a budget of $N = 5000$. The accuracy was tested on an independent test set of 5000 points, that were never used at any step in the algorithm. The parameters are set according to Section 5.1.

The algorithm was first challenged on three synthetic two-dimensional binary datasets (named dataset 1, 2, and 3, respectively), to study cases in which it is favorable. Dataset 1 aims at reproducing in two dimensions a toy example used by (Dasgupta, 2011), where the best linear classifier is located at $x_1 = -0.3$ but active learning algorithms could be misled to $x_1 = 0$. Dataset 2 represents a situation where some data ($x_1 < 0$) are easy to classify while others ($x_1 > 0$) are not. Dataset 3 is a mixture of Gaussian distributions, whose parameters can be adjusted to create various degrees of overlap. The results presented here correspond to $\sigma = 0.3$.

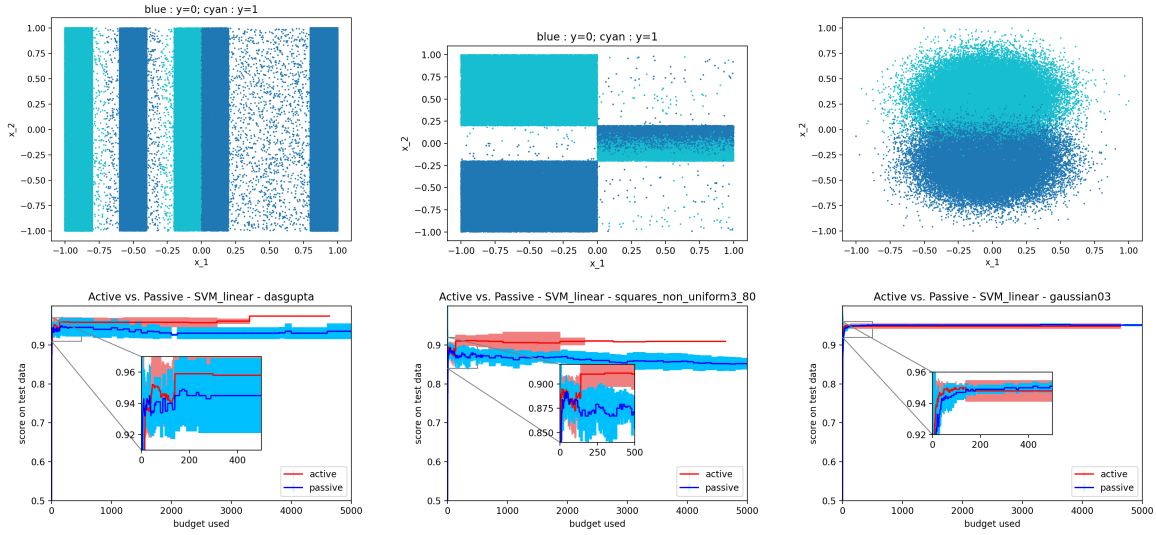

Figure 2: Top : From left to right, synthetic datasets 1, 2, and 3 used in this study with the points colored in blue or cyan depending on their class. Bottom : corresponding learning curves for active and passive linear classifiers.

The datasets are presented on Figure 2 as well as the corresponding learning curves for our active learning algorithm and its passive counterpart in the case of several classifiers: linear SVM, SVM with a Gaussian kernel, random forests and k nearest neighbors. These classifiers are from the `scikit-learn` library (Pedregosa *et al.*, 2011). Several parameters were tested, with similar results. The results in Table 1 are with the following parameters: regularization constant $C = 5$ for SVM, 100 trees for random forests, $k = 5$ for kNN. The other parameters are kept to their default value.

Results for datasets 1 and 2 In the case of SVM linear classifiers, our active learning algorithm is clearly superior to its passive counterpart for datasets 1 and 2, either with the larger budget ($N = 5000$) or with the smaller budget ($N = 200$). The situation is similar for SVM with Gaussian kernel, although it is less pronounced for dataset 2 at large budget. In the case of random forests and kNN, the difference is barely noticeable at large budget, but our algorithm is clearly superior with the smaller budget.

Results for dataset 3 Dataset 3 was designed to represent an easier classification problem. In this case our active learning algorithm does not present any advantage, although it does not significantly deteriorates the results (only slightly for SVM with Gaussian kernel).

5.3 Non-synthetic datasets

Several experiments were performed with various dataset from the UCI machine learning repository. Three "large" (more than 10000 data points) were used: *skin* (245057 points in $\mathbb{R}^3$), *fraud* (20468

dataset id	classifier	budget N	test precision	
			passive	active
1	SVM linear	5000	0.935 ± 0.020	**0.974 ± 0.00**
		200	0.945 ± 0.023	**0.959 ± 0.012**
	SVM rbf	5000	0.975 ± 0.003	**0.996 ± 0.002**
		200	0.964 ± 0.012	**0.989 ± 0.022**
	random forests	5000	1.000 ± 0.000	1.000 ± 0.000
		200	0.989 ± 0.004	**0.997 ± 0.008**
	kNN ($k = 5$)	5000	0.995 ± 0.002	0.995 ± 0.002
		200	0.956 ± 0.011	**0.993 ± 0.013**
2	SVM linear	5000	0.852 ± 0.015	**0.909 ± 0.000**
		200	0.871 ± 0.026	**0.910 ± 0.016**
	SVM rbf	5000	0.966 ± 0.003	**0.968 ± 0.003**
		200	0.951 ± 0.007	**0.967 ± 0.005**
	random forests	5000	0.965 ± 0.003	0.965 ± 0.003
		200	0.957 ± 0.005	**0.965 ± 0.003**
	kNN ($k = 5$)	5000	0.965 ± 0.003	**0.967 ± 0.003**
		200	0.950 ± 0.012	**0.963 ± 0.010**
3	SVM linear	5000	0.951 ± 0.003	0.948 ± 0.007
		200	0.949 ± 0.003	0.948 ± 0.007
	SVM rbf	5000	**0.952 ± 0.003**	0.943 ± 0.012
		200	**0.948 ± 0.003**	0.942 ± 0.011
	random forests	5000	0.944 ± 0.002	0.943 ± 0.006
		200	0.942 ± 0.008	0.943 ± 0.007
	kNN ($k = 5$)	5000	0.946 ± 0.004	0.945 ± 0.004
		200	0.944 ± 0.005	0.945 ± 0.003

Table 1: Results on synthetic datasets 1, 2, and 3 for budgets of 5000 or 200, with several classifiers: linear SVM, SVM with Gaussian kernel (called SVM rbf here), random forests (with 100 trees), and k nearest neighbors (kNN, with $k = 5$)

points in $\mathbb{R}^{113}$) and *EEG* (14980 points in $\mathbb{R}^{14}$). For those "large" datasets a maximum budget of $N = 3000$ was used.
Three "small" (less than 1000 data points) were also considered: *breast* (683 points in $\mathbb{R}^{10}$), *cleveland* (297 points in R^{13}), *credit* (690 points in R^{14}). For those "small" datasets a maximum budget of $N = 500$ was used.
The results for the largest dataset (*skin*) are presented as learning curves on Figure 3. All results are summarized in Table 2.

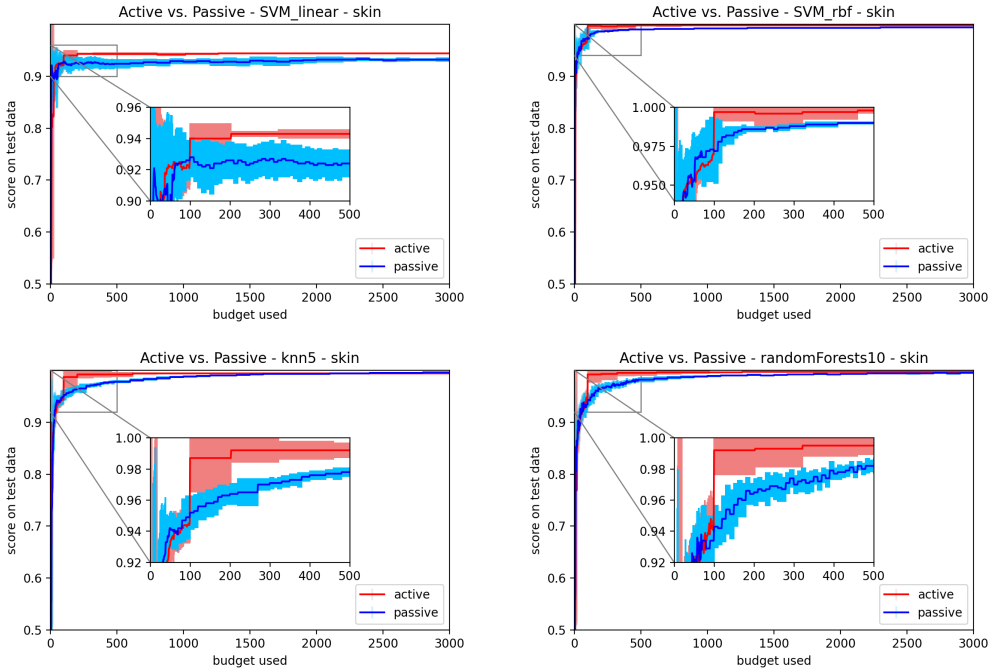

Figure 3: Skin dataset with linear SVM, rbf SVM, kNN5, and random forests: active vs. passive learning curves.

These results indicate that for the *skin* and *fraud* datasets, the converged accuracy (at large budget) is superior for active learning in the case of SVM linear, but very similar for the other classifiers. This is partially due to the fact that the resulting active classifier is not linear anymore. However, when the budget is limited to smaller values (see the inserts of Figure 3), the active learning procedure provides a clear advantage.
The picture remains unchanged when we consider the "small" datasets. Indeed, most of the time the active method improves the passive one (see Table 3). However, this improvement is rather limited, expect for *cleveland* dataset where the use of the active algorithm is particularly beneficial.

5.4 Summary of the results and discussion

The study on synthetic datasets shows that our active learning algorithm using rejection provides a clear advantage for the first two datasets, especially at low budget, but not for the third dataset. This indicates that our algorithm is most useful in situations where the classification problem is more difficult.
In non-synthetic datasets, the active learning procedure appears to be most effective on larger datasets. The explanation is as follows. For small datasets (*e.g.*, a few hundreds points), the number of points N_0 has to be chosen quite small. The estimate $\hat{\eta}_0$ is thus likely to be inaccurate, which in turn implies

name	classifier	passive	active
skin	SVM linear	0.931 ± 0.004	**0.944 ± 0.001**
	SVM rbf	0.994 ± 0.000	**0.998 ± 0.000**
	random forests	0.995 ± 0.001	**0.997 ± 0.000**
	kNN ($k = 5$)	**0.996 ± 0.000**	0.994 ± 0.000
fraud	SVM linear	0.994 ± 0.000	**0.999 ± 0.000**
	SVM rbf	0.988 ± 0.002	**0.993 ± 0.001**
	random forests	0.991 ± 0.006	**0.998 ± 0.002**
	kNN ($k = 5$)	0.946 ± 0.003	**0.959 ± 0.002**
EEG	SVM linear	**0.555 ± 0.005**	0.534 ± 0.000
	SVM rbf	0.549 ± 0.006	**0.559 ± 0.000**
	random forests	0.833 ± 0.005	**0.877 ± 0.028**
	kNN ($k = 5$)	**0.763 ± 0.007**	0.716 ± 0.009

Table 2: Results on "large" non-synthetic datasets with several classifiers for active and passive procedures, with a budget of $N = 3000$.

name	classifier	passive	active
breast	SVM linear	0.965 ± 0.008	**0.972 ± 0.006**
	SVM rbf	0.961 ± 0.008	**0.968 ± 0.011**
	random forests	0.968 ± 0.009	**0.970 ± 0.008**
	kNN ($k = 5$)	0.964 ± 0.008	0.965 ± 0.011
cleveland	SVM linear	**0.829 ± 0.047**	0.821 ± 0.011
	SVM rbf	0.804 ± 0.025	**0.906 ± 0.017**
	random forests	0.778 ± 0.029	**0.879 ± 0.059**
	kNN ($k = 5$)	0.797 ± 0.038	**0.815 ± 0.014**
credit	SVM linear	0.848 ± 0.023	0.847 ± 0.020
	SVM rbf	0.862 ± 0.017	0.851 ± 0.022
	random forests	0.845 ± 0.025	**0.853 ± 0.014**
	kNN ($k = 5$)	0.851 ± 0.024	**0.857 ± 0.019**

Table 3: Results on three "small" non-synthetic datasets with several classifiers and a budget not to exceed 500.

an inaccurate estimation of the uncertain region in the first steps and then leads to a poorly controlled algorithm.
Interestingly, even in such small datasets, our algorithm is rarely detrimental to the final precision reached and can even be useful when the budget is extremely limited.

6 Conclusion and perspectives

Recently several works have started to combine active learning and rejection arguments by abstaining to label some data within an active learning algorithm. This combination is very natural since active learning and rejection both focus on the most difficult data to classify. In this work, instead of completely abstaining to label some data, we use rejection principles in a novel way to estimate the uncertain region typically used in active learning algorithms. We therefore propose a computationally efficient active learning algorithm that combines active learning with rejection. We theoretically prove the merits of our algorithm and show through several numerical experiments that it can be efficiently applied to any off-the-shelf machine learning algorithm. The benefits are more pronounced when the label budget is limited, which is promising for practical applications.
Nevertheless, in the last steps of our algorithm the uncertainty about the label of some points can become very substantial, in which case it becomes natural to completely abstain from labeling. This abstention will be included in future work combined with our use of the reject option.

References

Audibert, J. & Tsybakov, A. (2007). Fast learning rates for plug-in classifiers. *Ann. Statist.* **35**, 608–633.

Balcan, M.-F., Beygelzimer, A. & Langford, J. (2009). Agnostic active learning. *Journal of Computer and System Sciences* **75**, 78–89.

Beygelzimer, A., Dasgupta, S. & Langford, J. (2009). Importance weighted active learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 49–56.

Castro, R. M. & Nowak, R. D. (2008). Minimax bounds for active learning. *IEEE Transactions on Information Theory* **54**, 2339–2353.

Chow, C. (1957). An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers* pp. 247–254.

Cohn, D., Atlas, L. & Ladner, R. (1994). Improving generalization with active learning. *Machine learning* **15**, 201–221.

Cortes, C., DeSalvo, G. & Mohri, M. (2016). Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pp. 67–82. Springer.

Dasgupta, S. (2011). Two faces of active learning. *Theoretical computer science* **412**, 1767–1781.

Dasgupta, S., Hsu, D. J. & Monteleoni, C. (2007). *A general agnostic active learning algorithm.* Citeseer.

Denis, C. & Hebiri, M. (2019). Consistency of plug-in confidence sets for classification in semi-supervised learning. *Journal of Nonparametric Statistics* .

Denis, C., Hebiri, M. & Zaoui, A. (2020). Regression with reject option and application to knn. *arXiv preprint arXiv:2006.16597* .

Devroye, L., Györfi, L. & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition.* Springer, New York.

Freund, Y., Seung, H. S., Shamir, E. & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine learning* **28**, 133–168.

Gadat, S., Klein, T. & Marteau, C. (2016). Classification in general finite dimensional spaces with the k-nearest neighbor rule. *The Annals of Statistics* **44**, 982–1009.

Giné, E. & Nickl, R. (2021). *Mathematical foundations of infinite-dimensional statistical models.* Cambridge university press.

Grandvalet, Y., Rakotomamonjy, A., Keshet, J. & Canu, S. (2009). Support vector machines with a reject option. In *NIPS*, pp. 537–544.

Hanneke, S. (2007). A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 353–360.

Hanneke, S. (2011). Rates of convergence in active learning. *The Annals of Statistics* pp. 333–361.

Hanneke, S. & Yang, L. (2015). Minimax analysis of active learning. *J. Mach. Learn. Res.* **16**, 3487–3602.

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

Herbei, R. & Wegkamp, M. (2006). Classification with reject option. *Canad. J. Statist.* **34**, 709–721.

Kpotufe, S., Yuan, G. & Zhao, Y. (2022). Nuances in margin conditions determine gains in active learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 8112–8126. PMLR.

Lei, J. (2014). Classification with confidence. *Biometrika* **101**, 755–769.

Locatelli, A., Carpentier, A. & Kpotufe, S. (2017). Adaptivity to noise parameters in nonparametric active learning. *Proceedings of Machine Learning Research vol* **65**, 1–34.

Locatelli, A., Carpentier, A. & Kpotufe, S. (2018). An adaptive strategy for active learning with smooth decision boundary. In *Algorithmic Learning Theory*, pp. 547–571. PMLR.

Lugosi, G. (2002). Pattern classification and learning theory. In *Principles of nonparametric learning*, pp. 1–56. Springer.

Massart, P. & Nédélec, É. (2006). Risk bounds for statistical learning. *Ann. Statist.* **34**, 2326–2366.

Minsker, S. (2012). Plug-in approach to active learning. *Journal of Machine Learning Research* **13**.

Naadeem, M., Zucker, J. & Hanczar, B. (2010). Accuracy-rejection curves (ARCs) for comparing classification methods with a reject option. In *MLSB*, pp. 65–81.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830.

Puchkin, N. & Zhivotovskiy, N. (2021). Exponential savings in agnostic active learning through abstention. In *Conference on Learning Theory*, pp. 3806–3832. PMLR.

Schreuder, N. & Chzhen, E. (2021). Classification with abstention but without disparities. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, vol. 161 of *Proceedings of Machine Learning Research*, pp. 1227–1236. AUAI Press.

Settles, B. (1994). Active learning literature survey. *Machine Learning* **15**, 201–221.

Shah, K. & Manwani, N. (2020). Online active learning of reject option classifiers. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**, 5652–5659.

Shekhar, S., Ghavamzadeh, M. & Javidi, T. (2021). Active learning for classification with abstention. *IEEE Journal on Selected Areas in Information Theory* **2**, 705–719.

Tsybakov, A. (2004). Optimal aggregation of classifiers in statistical learning. *Ann. Statist.* **32**, 135–166.

Urner, R., Wulff, S. & Ben-David, S. (2013). Plal: Cluster-based active learning. In *Conference on Learning Theory*, pp. 376–397. PMLR.

Vovk, V., Gammerman, A. & Saunders, C. (1999). Machine-learning applications of algorithmic randomness. In *In Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 444–453. Morgan Kaufmann.

Vovk, V., Gammerman, A. & Shafer, G. (2005). *Algorithmic learning in a random world*. Springer, New York.

Yuan, M. & Wegkamp, M. (2010). Classification methods with reject option based on convex risk minimization. *J. Mach. Learn. Res.* **11**, 111–130.

Zhu, Y. & Nowak, R. (2022). Efficient active learning with abstention. *arXiv preprint arXiv:2204.00043* .

Appendix

The section is devoted to the proof of the main result.

A Technical result

Let us first introduce some general notations: Let A be a subset of $[0,1]^d$, and a cubic partition $\mathcal{C}_r$ as introduced in Definition 3.4. For $R \in \mathcal{C}_r$, with $R \cap A \neq \emptyset$, we introduce the regression function in R as:

$$\bar{\eta}(R) = \frac{1}{\Pi(R)} \int_R \eta(z)\Pi(dz|A).$$

and we define $\bar{\eta}(x) := \bar{\eta}(R)$ for all $x \in R$.
Here, for each $k \geq 0$, and $r_k = N_k^{-1/(d+2)}$, we consider the estimator:

$$\hat{\eta}_k := \hat{\eta}_{A_k, \lfloor N_k \Pi(A_k) \rfloor, r_k}, \tag{A.1}$$

where $\hat{\eta}_{A_k, \lfloor N_k \Pi(A_k) \rfloor, r_k}$ is defined according to Definition 3.4, and A_k is defined in algorithm 1. importantly, defining $\hat{\eta}_k$ in this way for all $k \geq 0$ allows us to characterize the set A_k in an explicit form:

$$A_k = \bigcup_{R \in \mathcal{C}_{r_k},\, R \cap A_k \neq \emptyset} R.$$

We firstly provide a high probability bound on the estimation error:

Lemma A.1 (Favorable event with high probability)**.**
Let L be defined as:

$$L = \max\{j \geq 1, N > \sum_{k=0}^{j} \lfloor N_k \Pi(A_k) \rfloor\}. \tag{A.2}$$

Let $k \in \{0, \ldots, L\}$ and E be the event defined by:

$$E = \cap_{k=0}^{L} E_k, \tag{A.3}$$

where

$$E_k = \left\{ \| \eta - \hat{\eta}_k \|_{\infty, A_k} \leq c_5 \log\left(\frac{N}{\delta}\right) N_k^{-1/(2+d)} \right\}, \tag{A.4}$$

with $\| \eta - \hat{\eta}_k \|_{\infty, A_k} := \sup_{x \in A_k} |\hat{\eta}_k(x) - \eta(x)|$ and c_5 is a constant independent of N and N_k, but dependent on L and d. Under Assumptions 3.1 and 3.2 we have:

$$\mathbb{P}(E) \geq 1 - \delta.$$

Proof.
Let first note that L is deterministic as for all $k \geq 1$, $\Pi(A_k) = \varepsilon_k$, where ε_k is stated in our algorithm. Let $k \in \{0, \ldots, L\}$ and the corresponding estimator $\hat{\eta}_k$ (see (A.1)). Let $\mathcal{C}_{r_k}$ the cubic partition considered in Definition 3.4, and fix $R \in \mathcal{C}_{r_k}$. Let $x \in R$ with $R \cap A_k \neq \emptyset$.
Let $T_{j,k} = Y_j \mathbb{1}_{\{X_j \in R\}} \frac{\Pi(A_k)}{\Pi(R)}$. We observe that conditional to A_k, $\mathbb{E}[T_{j,k}] = \bar{\eta}(R)$, and

$$|T_{j,k}| \leq \frac{\Pi(A_k)}{\Pi(R)}. \tag{A.5}$$

Furthermore

$$\mathrm{Var}(T_{j,k}) = \frac{\Pi(A_k)^2}{\Pi^2(R)}\mathrm{Var}(Y_j \mathbb{1}_{\{X_j \in R\}}) \leq \frac{\Pi(A_k)}{\Pi^2(R)} \int_R \eta(z)\Pi(dz|A_k) \leq \frac{\Pi(A_k)}{\Pi(R)}. \tag{A.6}$$

Hence, from Bernstein Inequality, we deduce that for $t \leq 1$,

$$\mathbb{P}\left(|\hat{\eta}_k(x) - \bar{\eta}(R)| \geq t\right) \leq \exp\left(-\frac{\lfloor N_k \Pi(A_k)\rfloor t^2}{Var(T_{j,k}) + \frac{t\Pi(A_k)}{3\Pi(R)}}\right) \leq \exp\left(-\lfloor N_k\Pi(A_k)\rfloor \Pi(R) t^2/\Pi(A_k)\right)$$

by using (A.6).
Note that for $t > 1$, the inequality is always satisfied. Now, applying the above inequality, we deduce

$$\mathbb{P}\left(|\hat{\eta}(x) - \bar{\eta}(x)| \geq t\sqrt{\frac{\Pi(A_k)}{\lfloor N_k\Pi(A_k)\rfloor \Pi(R)}}\right) \leq \exp(-t^2),$$

Hence choosing $t = \sqrt{\log\left(\frac{N(L+1)}{c_1\delta}\right)}$, (where c_1 will be defined later) we deduce that for all $x \in R$, with probability at least $1 - \frac{c_1\delta}{N(L+1)}$, we have

$$|\hat{\eta}(x) - \bar{\eta}(x)| \leq \sqrt{\log\left(\frac{N(L+1)}{c_1\delta}\right)\frac{2}{N_k\Pi(R)}}$$

From the strong density assumption, we then obtain that for all $x \in R$, with probability at least $1 - \frac{c_1\delta}{N(L+1)}$,

$$|\hat{\eta}(x) - \bar{\eta}(x)| \leq c_2\sqrt{\log\left(\frac{N(L+1)}{c_1\delta}\right)\frac{1}{N_k r_k^d}}. \tag{A.7}$$

Where $c_2 = \sqrt{\frac{2}{c_1}}$, and c_1 is such that $\Pi(R) \geq c_1 r_k^d$ by Assumption 3.2.
To get a result in L_∞-norm on A_k, it remains to consider the union bound over all $R \in \mathcal{C}_{r_k}$ such that $R \cap A_k \neq \emptyset$.

$$\|\hat{\eta} - \bar{\eta}\|_{\infty, A_k} \leq \max_{R,\ R\cap A_k \neq \emptyset} \|\hat{\eta} - \bar{\eta}\|_{\infty, R}.$$

By definition, for all $k \geq 0$, the estimator $\hat{\eta}_k$ is constant on each cell R, in this case, we have:

$$\Pi(A_k) = \sum_{R,\ R\cap A_k = \emptyset} \Pi(R)$$

Then, by using Assumption 3.2, we have:

$$\Pi(A_k) \geq |\{R,\ R \cap A_k \neq \emptyset\}|\ c_1 r_k^d.$$

As $r_k = N_k^{-1/(d+2)}$, we get for all $k \in \{0, \ldots, L\}$,

$$|\{R,\ R \cap A_k \neq \emptyset\}| \leq \frac{1}{c_1}\Pi(A_k) N_k^{d/(d+2)} \leq \frac{1}{c_1}(\Pi(A_k)N_k) \leq \frac{N}{c_1} \tag{A.8}$$

Thus we have (conditional on A_k):

$$\mathbb{P}\left(\forall\, x \in A_k, |\hat{\eta}(x) - \bar{\eta}(x)| > c_2\sqrt{\frac{\log\left(\frac{N(L+1)}{c_1\delta}\right)}{N_k r_k^d}}\right) \leq \mathbb{P}\left(\max_{R,\, R\cap A_k\neq\emptyset} \|\hat{\eta} - \bar{\eta}\|_{\infty,R} > c_2\sqrt{\frac{\log\left(\frac{N(L+1)}{c_1\delta}\right)}{N_k r_k^d}}\right)$$
$$\leq \sum_{R,\, R\cap A_k=\emptyset} \mathbb{P}\left(\|\hat{\eta} - \bar{\eta}\|_{\infty,R} > c_2\sqrt{\frac{\log\left(\frac{N(L+1)}{c_1\delta}\right)}{N_k r_k^d}}\right)$$
$$\leq |\{R,\ R\cap A_k \neq \emptyset\}|\frac{c_1\delta}{N(L+1)}$$
$$\leq \frac{\delta}{L+1} \quad \text{by (A.8)}$$

Besides, Assumption 3.1 leads to

$$\|\eta - \bar{\eta}\|_{\infty,A_k} \leq c_3 r_k, \tag{A.9}$$

where c_3 depends on s (from Assumption 3.1) and d. Thus, by combining (A.7), (A.9) and (A.8), we can obtain that with probability at least $1 - \frac{\delta}{L+1}$,

$$\|\hat{\eta}_k - \eta\|_{\infty,A_k} \leq c_4\left(\sqrt{\log\left(\frac{N(L+1)}{c_1\delta}\right)\frac{1}{N_k r_k^d}} + r_k\right),$$

where $c_4 = \max(c_2, c_3)$.
Finally, as $r_k = N_k^{-1/2+d}$, by considering the union bound over all steps, we get with probability at least $1-\delta$,

$$\|\hat{\eta}_k - \eta\|_{\infty,A_k} \leq c_5 \log\left(\frac{N}{\delta}\right) N_k^{-1/(2+d)} \quad \text{for all } k \in \{0,\dots,L\} \tag{A.10}$$

where c_5 depends on c_4, c_1 and L. □

Because the constant c_5 in (A.4) depends on L, we provide below a result which states that the variable L defined in (A.2) does not affect drastically the bounds in (A.4).

Lemma A.2 (Bounds on the maximum number of steps L)**.**
Let us consider the variable L defined in (A.2)*, we have:*

$$\log_2\left(c_8\left(\frac{1}{\log\left(\frac{N}{\delta}\right)}\right)^{\frac{d+2}{1+d}} N^{\frac{d+3}{2+2d}}\right) \leq L$$

and

$$L \leq \min\left(1 + \log_2\left(\left(\frac{1}{c_6\log\left(\frac{N}{\delta}\right)}\right)^{(2+d)/(1+d)} N^{(3+d)/(2+2d)}\right), \log_2\left(\sqrt{N}\right)\right),$$

where c_8, c_6 are the constants respectively defined in (B.8)*, and* (B.2)*.*

Proof.
By definition of L, we have

$$N \leq \sum_{i=0}^{L+1} N_i \Pi(A_i)$$

and we have as in the proof of Lemma B.2

$$N_L \geq c_8 \left(\frac{1}{\log\left(\frac{N}{\delta}\right)} \right)^{(d+2)/(1+d)} N^{(d+2)/(1+d)}.$$

Besides, as $N_L = 2^L N_0$ and $N_0 = \sqrt{N}$, we obtain the first inequality

$$L \geq \log_2 \left(c_8 \left(\frac{1}{\log\left(\frac{N}{\delta}\right)} \right)^{(d+2)/(1+d)} N^{(d+3)/(2+2d)} \right) \tag{A.11}$$

We can get the second inequality by starting with (A.2), that is:

$$N_L \Pi(A_L) \leq N.$$

Furthermore, as $\Pi(A_L) = \varepsilon_L = \min\left(1, c_6 \log\left(\frac{N}{\delta}\right) N_{L-1}^{-1/(2+d)}\right)$ (see (B.3)), we get

$$N_L \min\left(1, c_6 \log\left(\frac{N}{\delta}\right) N_{L-1}^{-1/(2+d)}\right) \leq N.$$

If $1 \leq c_6 \log\left(\frac{N}{\delta}\right) N_{L-1}^{-1/(2+d)}$, then

$$L \leq \log_2\left(\sqrt{N}\right) \tag{A.12}$$

On the other hand, if $1 > c_6 \log\left(\frac{N}{\delta}\right) N_{L-1}^{-1/(2+d)}$ then

$$L \leq 1 + \log_2 \left(\left(\frac{1}{c_6 \log\left(\frac{N}{\delta}\right)} \right)^{(2+d)/(1+d)} N^{(3+d)/(2+2d)} \right) \tag{A.13}$$

Finally, by combining (A.11), (A.12), and (A.13), we get the second inequality.

□

B Proof of Theorem 3.5

We firstly prove that in the event E, the classifier $g_{\hat{\eta}_k}$ does not make any error of classification in the set $A_k \setminus A_{k+1}$ for all $k = 0, \ldots, L-1$, where L is defined by (A.2).

Lemma B.1 (Correct classification)**.**
Let E be the event defined by (A.3)*. Under Assumption 3.3, the Bayes classifier g^* agrees with $g_{\hat{\eta}_k}$ on the set $A_k \setminus A_{k+1}$ for $k \in \{0, \ldots, L-1\}$, where L is defined by* (A.2)*, and $\hat{\eta}_k$ by* (A.1)*.*

Proof.
Let us start by stating general facts that hold for a generic estimator $\hat{\eta}$ and the corresponding score function $\hat{f}(x) = \max(\hat{\eta}(x), 1 - \hat{\eta}(x))$. We consider F_f, and $F_{\hat{f}}$ the cumulative distribution of $f(X)$ and $\hat{f}(X)$, where $f(x) = \max(\eta(x), 1 - \eta(x))$. Let $t \in (1/2, 1)$, we have that conditional on the data

$$F_{\hat{f}}(t) \leq \left| F_{\hat{f}}(t) - F_f(t) \right| + F_f(t).$$

Besides, the following relation holds:

$$\left| F_{\hat{f}}(t) - F_f(t) \right| \leq \mathbb{E}_X \left[\mathbb{1}_{\{\|\hat{f} - f\|_\infty \geq |f(X) - t|\}} \right] \leq 2C \|\hat{f} - f\|_\infty,$$

where C is the bound on the density f provided in Assumption 3.3. Using again Assumption 3.3 we can write

$$F_f(t) \leq C\left(t - \frac{1}{2}\right).$$

We then deduce that for all $t \in (1/2, 1)$, conditional on the data

$$F_{\hat{f}}(t) \leq 2C\|\hat{f} - f\|_\infty + C\left(t - \frac{1}{2}\right) \leq 2C\|\hat{\eta} - \eta\|_\infty + C\left(t - \frac{1}{2}\right). \tag{B.1}$$

Given iteration $k \in \{0, \ldots, L-1\}$, we set $\hat{t}_k = \|\hat{\eta}_k - \eta\|_{\infty, A_k}$, and $t_k = \frac{1}{2} + \hat{t}_k$. Thanks to (B.1), with $\hat{\eta} = \hat{\eta}_k$ and $t = t_k$, we deduce that (conditional on A_k)

$$F_{\hat{f}_k}(t_k) \leq 3C\hat{t}_k.$$

Then, in the event E, we have that

$$F_{\hat{f}_k}(t_k) \leq c_6 \log\left(\frac{N}{\delta}\right) N_k^{-1/(2+d)}, \tag{B.2}$$

where $c_6 = 3c_5C$, and c_5 is defined in (A.10). Hence,

$$F_{\hat{f}_k}(t_k) \leq \min\left(1, c_6 \log\left(\frac{N}{\delta}\right) N_k^{-1/(2+d)}\right) \leq \varepsilon_{k+1} \tag{B.3}$$

This implies that $\lambda_{k+1} \geq \frac{1}{2} + \hat{t}_k$ by the definition of of λ_{k+1}.
Let $x \in A_k \setminus A_{k+1} = \{x \in A_k,\ \ \hat{f}_k(x) > \lambda_{k+1}\}$. Necessarily, we have

$$\hat{f}_k(x) - \frac{1}{2} > \|\hat{\eta}_k - \eta\|_{\infty, A_k} \geq |\hat{\eta}_k(x) - \eta(x)|$$

which implies $g_\eta(x) = g_{\hat{\eta}_k}(x)$. □

Lemma B.2 (Excess-error)**.**
Let $g_{\hat{\eta}}$ be the classifier provided by our algorithm, on the event E, we have

$$R(g_{\hat{\eta}}) - R(g_\eta) \leq \tilde{O}\left(N^{-\frac{2}{d+1}}\right),$$

where $\tilde{O}$ hides some constants and logarithmic factors.

Proof. Let us consider the sequence $(A_k)_{0 \leq k \leq L}$ used in our algorithm. It is not difficult to see that $\{A_k \setminus A_{k+1}, k = 0, \ldots, L-1\} \cup A_L$ forms a partition of $[0,1]^d$, where L is defined by (A.2).
In this case, the excess-risk of $g_{\hat{\eta}}$ can be rewritten as:

$$R(g_{\hat{\eta}}) - R(g^*) = \sum_{j=0}^{L-1} \int_{\{g_{\hat{\eta}} \neq g^*\} \cap \{A_j \setminus A_{j+1}\}} |2\eta(x) - 1| d\Pi(x) + \int_{A_L \cap \{g_{\hat{\eta}} \neq g^*\}} |2\eta(x) - 1| d\Pi(x)$$

and thus

$$R(g_{\hat{\eta}}) - R(g^*) = 2\sum_{j=1}^{L-1} \mathbb{E}_X\left[|\eta(X) - \frac{1}{2}| \mathbb{1}_{\{g^*(X) \neq g_{\hat{\eta}_j}(X)\}} \mathbb{1}_{\{A_j \setminus A_{j+1}\}}\right] + 2\mathbb{E}_X\left[|\eta(X) - \frac{1}{2}| \mathbb{1}_{\{g^*(X) \neq g_{\hat{\eta}_L}(X)\}} \mathbb{1}_{\{A_L\}}\right] \tag{B.4}$$

Due to the Lemma B.1, the first term in the r.h.s of (B.4) is zero in the event E. Thus we get

$$\begin{aligned} R(\hat{g}) - R(g^*) &= 2\mathbb{E}_X \left[|\eta(X) - \frac{1}{2}| \mathbb{1}_{\{g^*(X) \neq g_{\hat{\eta}_L}(X)\}} \mathbb{1}_{\{A_L\}} \right] \\ &\leq 2\mathbb{E}_X \left[|\eta(X) - \frac{1}{2}| \mathbb{1}_{|\hat{\eta}(X) - \frac{1}{2}| < |\hat{\eta}_L(X) - \eta(X)|} \mathbb{1}_{\{A_L\}} \right] \end{aligned}$$

We thus have

$$\begin{aligned} R(\hat{g}) - R(g^*) &\leq 2\|\hat{\eta}_L - \eta\|_{\infty, A_L} . \mathbb{E}_X \left[\mathbb{1}_{|\hat{\eta}(X) - \frac{1}{2}| < |\hat{\eta}_L(X) - \eta(X)|} \right] \\ &\leq 4C \|\hat{\eta}_L - \eta\|^2_{\infty, A_L} \text{ by Assumption 3.3.} \end{aligned} \tag{B.5}$$

By Lemma A.1, we get with probability at least $1 - \delta$

$$R(\hat{g}) - R(g^*) \leq 4C c_6 \log^2 \left(\frac{N}{\delta} \right) N_L^{-2/(2+d)}. \tag{B.6}$$

Besides, because of the geometric progression of N_j, and the definition of L, we have

$$\begin{aligned} N &\leq \sum_{j=0}^{L+1} N_j \Pi(A_j) \\ &= \sum_{j=0}^{L+1} N_j \varepsilon_j \\ &\leq N_0 + c_6 \log \left(\frac{N}{\delta} \right) \sum_{j=1}^{L+1} N_j N_{j-1}^{-1/(2+d)} \\ &= N_0 + 2c_6 \log \left(\frac{N}{\delta} \right) \sum_{j=1}^{L+1} N_{j-1}^{(d+1)/(2+d)} \\ &\leq N_0 + c_7 \log \left(\frac{N}{\delta} \right) N_{L+1}^{(d+1)/(2+d)} \quad \text{for some constant c7.} \end{aligned}$$

Thus we get

$$\begin{aligned} N - N_0 \leq c_7 \log \left(\frac{N}{\delta} \right) N_{L+1}^{(d+1)/(2+d)} &\implies \frac{1}{4} N \leq c_7 \log \left(\frac{N}{\delta} \right) N_{L+1}^{(d+1)/(2+d)} \quad \text{as } N_0 = \sqrt{N} \leq \frac{3}{4} N \\ &\implies N_L \geq c_8 \left(\frac{1}{\log \left(\frac{N}{\delta} \right)} \right)^{(d+2)/(1+d)} N^{(d+2)/(1+d)}, \end{aligned} \tag{B.7}$$

where

$$c_8 = \frac{1}{2} \left(\frac{1}{4c_7} \right)^{(d+2)/(1+d)}. \tag{B.8}$$

Thus, (B.6) becomes

$$R(g_{\hat{\eta}}) - R(g_\eta) \leq \tilde{O} \left(N^{-\frac{2}{d+1}} \right).$$

□