Journal of Computer and System Sciences 78 (2012) 1481–1495

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Reliable agnostic learning

Adam Tauman Kalai [a,1], Varun Kanade [b,*,2], Yishay Mansour [c,3]

[a] Microsoft Research, Cambridge, MA, USA
[b] School of Engineering and Applied Sciences, Harvard University, Cambridge, MA, USA
[c] School of Computer Science, Tel Aviv University, Tel Aviv, Israel

ARTICLE INFO

Article history:
Received 20 January 2010
Received in revised form 12 February 2011
Accepted 22 December 2011
Available online 18 January 2012

Keywords:
PAC learning
Classification
Agnostic learning

ABSTRACT

It is well known that in many applications erroneous predictions of one type or another must be avoided. In some applications, like spam detection, false positive errors are serious problems. In other applications, like medical diagnosis, abstaining from making a prediction may be more desirable than making an incorrect prediction. In this paper we consider different types of *reliable classifiers* suited for such situations. We formalize the notion and study properties of reliable classifiers in the spirit of agnostic learning (Haussler, 1992; Kearns, Schapire, and Sellie, 1994), a PAC-like model where no assumption is made on the function being learned. We then give two algorithms for reliable agnostic learning under natural distributions. The first reliably learns DNFs with no false positives using membership queries. The second reliably learns halfspaces from random examples with no false positives or false negatives, but the classifier sometimes abstains from making predictions.

1. Introduction

In many machine learning applications, a crucial requirement is that mistakes of one type or another should be avoided at all cost. As a motivating example, consider spam detection, where classifying a correct email as spam (a false positive) can have dire, or even fatal consequences. On the other hand, if a spam message is not tagged correctly (a false negative) it is comparatively a smaller problem. In other situations, abstaining from making a prediction might be better than making wrong predictions; in a medical test it is preferable to have inconclusive predictions to wrong ones, so that the remaining predictions can be relied upon. A reliable classifier would have a pre-specified error bound for false positives or false negatives or both. We present formal models for different types of *reliable classifiers* and give efficient algorithms for some of the learning problems that arise.

In agnostic learning [9,13], one would like provably efficient learning algorithms that make no assumption about the target function being learned. The learner's goal is to nearly match (within ϵ) the accuracy of the best classifier from a specified class of functions (see Fig. 1(a)). Agnostic learning can be viewed as PAC learning [18] with arbitrary noise. In reliable agnostic learning, the learner's goal is to output a nearly reliable classifier whose accuracy nearly matches the accuracy of the best *reliable* classifier from a specified class of functions (see Fig. 1(b)–(c)).

* Corresponding author.
E-mail addresses: adum@microsoft.com (A.T. Kalai), vkanade@fas.harvard.edu (V. Kanade), mansour@tau.ac.il (Y. Mansour).

[1] Part of this research was done while the author was at Georgia Institute of Technology, supported in part by NSF SES-0734780, and NSF CAREER award, and a SLOAN Fellowship.
[2] This research was done while the author was at Georgia Institute of Technology, supported in part by NSF CCF-0746550.
[3] This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, by a grant from the Israel Science Foundation and by a grant from United States–Israel Binational Science Foundation (BSF). This publication reflects the authors' views only.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2011.12.026

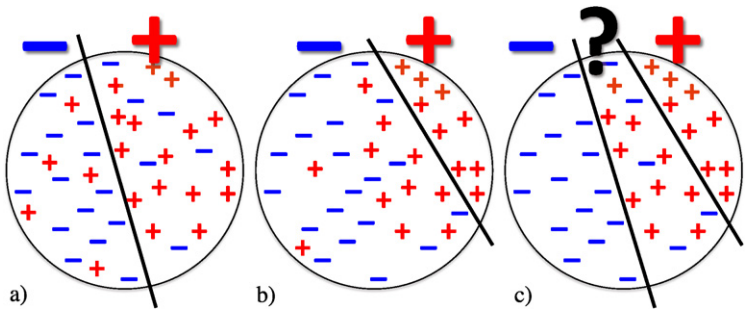

Fig. 1. Three different learning settings. Depending on the application and data, one may be more appropriate than the others. (a) The best (most accurate) classifier (from the class of halfspaces) for a typical agnostic learning problem. (b) The best positive-reliable halfspace classifier for a problem like spam-prediction in which false positives are to be avoided. (c) The best fully-reliable halfspace sandwich classifier for a problem in which all errors are to be avoided. It predicts $-$, $+$, or ?.

Since agnostic learning is extremely computationally demanding, it is interesting that we can find efficient algorithms for reliably agnostic learning interesting classes of functions over natural distributions. Our algorithms build on recent results in agnostic learning, one for learning decision trees [8] and one for learning halfspaces [12]. Throughout the paper, our focus is on *computationally efficient* algorithms for learning problems. The contributions of this paper are the following:

- We introduce models for reliable agnostic learning. Following prior work on agnostic learning, we consider both distribution-specific and distribution-free learning, as well as both learning from membership queries and from random examples. We show that reliable agnostic learning is no harder than agnostic learning and no easier that PAC learning.
- We give an algorithm for reliably learning DNF (with almost no false positives) over the uniform distribution, using membership queries. More generally, we show that if concept class C is agnostically learnable, then the class of disjunctions of concepts from C is reliably learnable (with almost no false positives). The dual statement – if C is agnostically learnable, then the class of conjunctions of concepts from C is reliably learnable (with almost no false negatives) – also holds.
- We give an algorithm for reliably learning halfspace sandwich classifiers over the unit ball under uniform distribution. This algorithm is fully-reliable in the sense that it almost never makes mistakes of any type, but may sometimes predict unknown ("?") as label. We also extend this algorithm to have *tolerant reliability*, in which case a permissible rate of false positives and false negatives is specified, and the goal of the algorithm is to achieve maximal accuracy subject to these constraints.

1.1. Positive reliability

A *positive-reliable* classifier is one that makes no false positive errors. (One can similarly define a *negative-reliable* classifier as one that makes no false negative errors.) For the spam example, a positive-reliable DNF expression could be

$$(\texttt{Nigeria} \wedge \texttt{bank} \wedge \texttt{transaction} \wedge \texttt{million}) \vee (\texttt{Viagra} \wedge \neg \texttt{COLT} \ldots.$$

An email should be almost certainly spam if it fits into any of a number of categories, where each category is specified by a sets of words that it must contain and must not contain.

Our goal is to output a classifier that is (almost) positive-reliable, i.e. has a very low rate of false positives (at most ϵ) and has false negative rate (almost) as low as the best positive-reliable classifier from a fixed set of classifiers, such as s-term DNFs. The best positive-reliable classifier is the one that has the least rate of false negatives; we require our algorithm to output a classifier that has rate of false negatives (within ϵ) as low as this best one, and require that the false positive rate of our classifier be less than ϵ.

Our first algorithm efficiently learns the class of DNF expressions in the positive-reliable agnostic learning model over the uniform distribution and uses membership queries. Our model is conceptually akin to a one-sided error agnostic noise model; hence this may be a step towards agnostically learning DNF. Note that our algorithm also gives an alternate way of learning DNFs (without noise) that is somewhat different than Jackson's celebrated harmonic sieve [10].

More generally, we show that if a concept class C is efficiently agnostically learnable, then polynomial-sized disjunctions of concepts from C are efficiently learnable in the positive-reliable agnostic model. Similarly, it can be shown that agnostically learning a concept class C implies learning polynomial-sized conjunctions of concepts from C in the negative-reliable agnostic model. In the reductions, we repeatedly make calls to the black-box agnostic learner for C, but we never change the distribution over unlabeled examples passed to this learner. Instead, we flip labels of examples in a manner that gives us an (almost) positive-reliable hypothesis allowing the reductions to be applied to distribution-specific agnostic learning algorithms. A consequence of this is that a polynomial-time algorithm for agnostically learning DNFs under the uniform distribution (and hence also CNFs) would give an efficient algorithm for the challenging problem of PAC learning depth-3

circuits over the uniform distribution (since PAC learning is easier than reliable learning). This idea of flipping labels, while leaving the distribution over unlabeled examples, has also been used to obtain distribution-specific boosting algorithms in the agnostic setting (see Kalai and Kanade [11] and Feldman [4]).

1.2. Full reliability

We consider the notion of *full reliability*, which means simultaneous positive and negative reliability. In order to achieve this, we need to consider *partial classifiers*, ones that may predict positive, negative, or "?", where a "?" means no prediction. A partial classifier is fully-reliable if it never produces false positives or false negatives. The accuracy of a partial classifier is the probability that it makes a correct prediction (different from "?") and the best fully-reliable partial classifier is one with maximum accuracy. Given a concept class of partial classifiers, the goal is to find a (nearly) fully-reliable classifier that is almost as accurate as the best fully-reliable classifier from the concept class.

We show that reliable agnostic learning is easier (or no harder) than agnostic learning; if a concept class is efficiently learnable in the agnostic setting, it is also efficiently learnable in the positive-reliable, negative-reliable, and full-reliable models.

1.3. Tolerant reliability

We also consider the notion of tolerant reliability, where we are willing to tolerate given rates τ_+, τ_- of false positives and false negatives. In this most general version, we consider the class of halfspace sandwich partial classifiers (halfspace sandwiches for short), in which the examples that are in one halfspace are positive, examples that are in another halfspace are negative, and the rest of the examples are classified as "?". We extend the agnostic halfspace algorithm and analysis of Kalai, Klivans, Mansour, and Servedio [12] to the case of halfspace sandwiches. In particular, we show that given arbitrary rates τ_+, τ_-, we can learn the class of halfspace sandwiches to within any constant ϵ under the uniform distribution over the unit ball in n dimensions in polynomial time using random examples. Our algorithm outputs a hypothesis h such that h has false positive and false negative rates close (within ϵ) to τ_+ and τ_- respectively and has accuracy within ϵ of the best halfspace sandwich with false positive and false negative rates bounded by τ_+, τ_-.

1.4. Related work

A classical approach to the problem of reliable classification is using a loss function that has different penalties for different types of errors – the *cost-sensitive classification* model (see [2,3]). For example, by having an infinite loss on false positive errors and a loss of one on false negative errors, we can essentially define a positive-reliable classifier as one that minimizes the loss. The idea of balancing the training set to account for different costs for different types of mistakes has been proposed [3]. The main issue that arises is computational, since there is no efficient way to compute a hypothesis that would minimize a general loss function. Even when the mistake costs are equal, the problem of agnostically learning DNF remains an important open problem [8]. The focus of this paper is providing polynomial-time algorithms for some interesting classes with theoretical guarantees in the agnostic setting. Another important point of distinction between this work and previous approaches suggested is that we do not change the distribution over unlabeled examples when using black-box learners, but instead flip labels, allowing our reductions to be applied to distribution-specific learning algorithms. This is important since most known agnostic learning algorithms are distribution-specific.

Conceptually, our work is similar to the study of cautious classifiers, i.e. classifiers which may output "unknown" as a label [6]. The models we introduce are similar to those considered earlier in literature – for example the full-reliable learning model is very similar to the bounded-improvement model in [16]. The methods we use to prove the reduction between agnostic learning and reliable learning are somewhat related to work on delegating classifiers [5] and cost-sensitive learning [3,19].

The motivation of our work is highly related to the Neyman–Pearson criterion, an application of which is the following: given a joint distribution over points and labels, minimize the rate of false negatives subject to the condition that the rate of false positives is bounded by a given input parameter. The Neyman–Pearson criterion classifies points based on the ratio between the likelihood of the point being labeled positive and negative. Neyman and Pearson [15] proved that the optimal classification strategy is to choose a threshold on the ratio of the likelihoods. This method was applied to statistical learning in [1,17], to solve constrained versions of empirical risk minimization or structural risk minimization. Unfortunately, the optimization problems that arise for most classes of interest are computationally intractable. In contrast, our work focuses on deriving computationally efficient algorithms for some interesting concept classes.

The focus of our work is to develop formal models for reliable learning and give algorithms with theoretical guarantees on their performance. Our models are similar in spirit to Valiant's PAC learning model [18] and agnostic learning models by Kearns, Schapire and Sellie [13]. We provide a way to use distribution-specific agnostic learning algorithms to obtain efficient algorithms for several different models of reliable learning. In this paper we focus on polynomial-time algorithms, however our reductions are applicable even if the learning algorithms are super-polynomial. For example, our results imply a positive-reliable algorithm for learning DNF using random examples in time $O(n^{\log(n)})$ under uniform distribution using the fact that conjunctions can be agnostically learned in $O(n^{\log(n)})$ time.

1.5. Organization

For readability, the paper is divided into three parts. In Section 3, we focus only on positive reliability. Section 4 contains a reduction from agnostic learning to fully-reliable learning. Finally, in Section 5, we consider reliable learning with pre-specified permissible error rates.

2. Preliminaries

Throughout the paper, we will consider $\langle X_n\rangle_{n\geqslant 1}$ as the instance space (for example the boolean cube $X_n = \{0,1\}^n$ or the unit ball $X_n = \mathcal{B}_n$), with distributions $\langle \mu_n\rangle_{n\geqslant 1}$ over the unlabeled examples. For each n, the target is an arbitrary function $f_n : X_n \to [0,1]$, which we interpret as $f_n(x) = \Pr[y = 1 \mid x]$. The distribution μ_n over X_n and the target function $f_n : X_n \to [0,1]$ together induce a joint distribution $D_n = (\mu_n, f_n)$ over labeled examples $X_n \times \{0,1\}$ as follows: To draw a random example (x, y) from D_n, pick $x \in X_n$ according to distribution μ_n and set $y = 1$ with probability $f_n(x)$, otherwise set $y = 0$. Throughout the paper, we will use μ to denote the distribution over unlabeled examples, and $D = (\mu, f)$ to denote the joint distribution over examples and labels.

A *false positive* is a prediction of 1 when the label is $y = 0$. Similarly a *false negative* is a prediction of 0 when the label is $y = 1$. The rate of false positives and negatives of a classifier $c : X_n \to \{0,1\}$ are defined below. When the distribution $D_n = (\mu_n, f_n)$ is clear from context, we will omit it.

$$\operatorname{false}_+(c, D_n) \triangleq \Pr_{(x,y)\sim D_n}\big[c(x) = 1 \wedge y = 0\big] = \mathop{\mathbf{E}}_{x\sim\mu_n}\big[c(x)\big(1 - f_n(x)\big)\big],$$
$$\operatorname{false}_-(c, D_n) \triangleq \Pr_{(x,y)\sim D_n}\big[c(x) = 0 \wedge y = 1\big] = \mathop{\mathbf{E}}_{x\sim\mu_n}\big[\big(1 - c(x)\big)f_n(x)\big].$$

A classifier c is said to be *positive-reliable* if $\operatorname{false}_+(c) = 0$. In other words, it never makes false positive predictions. Although, we focus on positive reliability, an entirely similar definition can be made in terms of negative reliability. The *error* of classifier c is

$$\operatorname{err}(c, D_n) \triangleq \Pr_{(x,y)\sim D_n}\big[c(x) \neq y\big] = \operatorname{false}_+(c) + \operatorname{false}_-(c).$$

To keep notation simple, we will drop the subscript n, except in definitions.

2.1. Oracles

As is standard in computational learning theory, we consider two types of oracles: membership query (MQ) and example (EX). Given a target function $f : X \to [0,1]$ we define the behavior of the two types of oracles below. In the case of example oracle, we assume that there is an underlying distribution μ over the unlabeled examples from X, and the examples are drawn from $D = (\mu, f)$.

- *Membership Query (MQ) oracle*: For a query $x \in X$, the oracle returns $y = 1$ with probability $f(x)$, and $y = 0$ otherwise, independently each time it is invoked.
- *Example (EX) oracle*: When invoked, the oracle draws $x \in X$ according to the distribution μ on X, sets $y = 1$ with probability $f(x)$, $y = 0$ otherwise and returns (x, y). Thus (x, y) is returned according to joint distribution $D = (\mu, f)$.

The reductions presented in Sections 3 and 4 work with respect to both types of oracles. Thus the reliable learning algorithms we provide have the same properties as that of the black-box agnostic learning algorithm used – for example, if the black-box learning algorithm is a membership query algorithm, our learning algorithm will be a membership query algorithm as well.

We employ the notation $\mathcal{O}(f)$ to denote an oracle, which may be either an example oracle or a membership query oracle. Note that neither type of oracle we consider is persistent – either may return $(x, 1)$ and $(x, 0)$ for the same x. In this sense, our model is similar to the p-concept distribution model for agnostic learning introduced in Kearns, Schapire and Sellie [13]. Whenever an algorithm $\mathcal{A}$ has access to oracle $\mathcal{O}(f)$ we denote it by $\mathcal{A}^{\mathcal{O}(f)}$.

We begin by defining agnostic learning and later give similar definitions for reliable learning.

Agnostic learning. Algorithm $\mathcal{A}$ *efficiently agnostically learns* sequence of concept classes $\langle C_n\rangle_{n\geqslant 1}$, under distributions $\langle \mu_n\rangle_{n\geqslant 1}$ over unlabeled examples, if there exists a polynomial $p(n, 1/\epsilon, 1/\delta)$ such that, for every $n \geqslant 1$, $\epsilon, \delta > 0$ and every $f_n : X_n \to [0,1]$ (inducing $D_n = (\mu_n, f_n)$), with probability at least $1 - \delta$, $\mathcal{A}^{\mathcal{O}(f_n)}(\epsilon, \delta)$ outputs hypothesis h that satisfies

$$\operatorname{err}(h, D_n) \leqslant \min_{c\in C_n} \operatorname{err}(c, D_n) + \epsilon.$$

The time complexity of both $\mathcal{A}$ and h is bounded by $p(n, 1/\epsilon, 1/\delta)$.

3. Positive reliability

In this section, we first formally define the notion of positive-reliable learning and then prove the following result: if a concept class C is efficiently agnostically learnable, the class of size-s disjunctions of concepts from C is efficiently positive-reliably learnable. An application of this result is an algorithm for positive-reliably learning s-term DNF expressions, based on the recent result by Gopalan, Kalai and Klivans [8] for agnostically learning decision trees. A conjunction has a small decision tree and hence the class of s-term DNF expressions is included in the class of size-s disjunctions of decision trees.

Below, we define positive-reliable learning and postpone definitions of full reliability and tolerant reliability to Sections 4 and 5 respectively. Given a class of classifiers C, define the subset of positive-reliable classifiers (relative to a fixed distribution and target function), to be $C^+ = \{c \in C \mid \text{false}_+(c) = 0\}$. Note that if we assume that C contains the classifier which predicts 0 on all examples, then C^+ is non-empty.

Positive-reliable learning. Algorithm $\mathcal{A}$ *efficiently positive-reliably learns* a sequence of concept classes $\langle C_n \rangle_{n \geqslant 1}$, under distributions $\langle \mu_n \rangle_{n \geqslant 1}$ over unlabeled examples, if there exists a polynomial $p(n, 1/\epsilon, 1/\delta)$ such that, for every $n \geqslant 1$, $\epsilon, \delta > 0$ and every $f_n : X_n \to [0,1]$ (inducing $D_n = (\mu_n, f_n)$), with probability at least $1 - \delta$, $\mathcal{A}^{\mathcal{O}(f_n)}(\epsilon, \delta)$ outputs hypothesis h that satisfies $\text{false}_+(h, D_n) \leqslant \epsilon$ and

$$\text{false}_-(h, D_n) \leqslant \min_{c \in C_n^+} \text{false}_-(c, D_n) + \epsilon.$$

The time complexity of both $\mathcal{A}$ and h is bounded by $p(n, 1/\epsilon, 1/\delta)$.

Each oracle call is assumed to take unit time. Hence, an upper bound on the run-time is also an upper bound on the sample complexity, i.e. the number of oracle calls.

3.1. Positive-reliably learning DNF

In this section, we prove that if a concept class C is efficiently agnostically learnable, then the class of size-s disjunctions of concepts from C is efficiently positive-reliably learnable. We first show that if a concept class is efficiently agnostically learnable, it is also efficiently positive-reliably learnable. We then show that a simple boosting-like algorithm can be used to positive-reliably learn disjunctions of concepts from this class. Below is an informal statement of the main theorem, which we shall state formally and prove in this section.

Theorem 1. *Let $\mathcal{A}$ be an algorithm that (using oracle $\mathcal{O}(f)$) efficiently agnostically learns concept class C under distribution μ. There exists an algorithm $\mathcal{A}'$ that (using oracle $\mathcal{O}(f)$ and black-box access to $\mathcal{A}$) efficiently positive-reliably learns the class of size-s disjunctions of concepts from C.*

Using the result on learning decision trees [8], we immediately get Corollary 1. The remainder of this section is devoted to proving Theorem 1 and Corollary 1.

Corollary 1. *There is an MQ algorithm $\mathcal{B}$ such that for any $n, s \geqslant 1$, $\mathcal{B}$ positive-reliably learns the class of s-term DNF expressions on n variables in time $\text{poly}(n, s, \frac{1}{\epsilon}, \frac{1}{\delta})$ with respect to the uniform distribution.*

We first prove the following simple, but very useful lemma: if we have access to oracle $\mathcal{O}(f)$, we can simulate oracle $\mathcal{O}(f')$ where $f'(x) = q(x)f(x) + (1 - q(x))r(x)$, where $q, r : X \to [0,1]$ are arbitrary functions and we only assume black-box access to q and r.

Lemma 1. *Given access to oracle $\mathcal{O}(f)$ and black-box access to the functions $q, r : X \to [0,1]$, we can simulate oracle $\mathcal{O}(f')$ for $f' = qf + (1-q)r$.*

Proof. We show this assuming that $\mathcal{O}(f)$ is an example oracle; the case when $\mathcal{O}(f)$ is a membership query oracle is simpler. Let $f' = qf + (1-q)r$. To simulate $\mathcal{O}(f')$, we do the following: First draw (x, y) from $\mathcal{O}(f)$, with probability $q(x)$, return (x, y). With probability $1 - q(x)$ do the following: Set $y' = 1$ with probability $r(x)$, $y' = 0$ otherwise, return (x, y'). It is easy to see that $\Pr[y = 1 \mid x] = f'(x)$, thus this simulates the oracle $\mathcal{O}(f')$. □

Using Lemma 1, we can simulate a different oracle and use a black-box agnostic learner to obtain a positive-reliable learner. The distribution μ over unlabeled examples is unchanged, but instead the target f is modified in a manner such that false positives (with respect to the original function) are penalized much more than false negatives. By an appropriate choice of parameters, the output of the black-box agnostic learner is close to the best positive-reliable classifier.

A crucial requirement of the Kushilevitz–Mansour [14,7] algorithm for learning parities (and hence also of the many other algorithms that use this as a black-box) is that the distribution over unlabeled examples be uniform (or product). Other known algorithms (e.g. learning halfspaces [12]) also have similar requirements that the distribution over unlabeled examples be from a certain class (e.g. normal or log-concave). While most reductions in learning theory change distributions, the ability of agnostic learning algorithms to handle arbitrary target functions makes possible for reductions that leave the distribution over unlabeled examples unchanged. Most of these algorithms are also robust to non-persistent labels, i.e. the same instance seen with different labels (cf. Appendix A of [8]).

We define an algorithm PRL (positive-reliable learner) with the following properties. The algorithm PRL takes as input the following:

- ϵ – the accuracy parameter,
- δ – the confidence parameter,
- $\mathcal{O}(f)$ – oracle access to the target function.

It uses black-box access to the agnostic learning algorithm $\mathcal{A}$ for class C under distribution μ. For target function f, it then defines $f' = (\frac{1}{2} + \frac{\epsilon}{4})f$, and makes a call to the black-box agnostic learning algorithm $\mathcal{A}$ simulating oracle $\mathcal{O}(f')$ and with parameters $\epsilon^2/2$ and δ. The hypothesis h returned, satisfies $\text{false}_+(h) \leqslant \epsilon$ and $\text{false}_-(h) \leqslant \min_{c \in C^+} \text{false}_-(c) + \epsilon$. Lemma 2 proves this claim.

Lemma 2. *Assume that algorithm $\mathcal{A}^{\mathcal{O}(f)}(\epsilon, \delta)$ efficiently agnostically learns concept class C under distribution μ (over X) in time $T(\epsilon, \delta)$. Then, algorithm* PRL *which returns the value of $\mathcal{A}^{\mathcal{O}(f')}(\epsilon^2/2, \delta)$, where $f' = (1/2 + \epsilon/4) f$, positive-reliably learns C under μ in time $T(\epsilon^2/2, \delta)$.*

Proof. Let $f' = (\frac{1}{2} + \frac{\epsilon}{4})f$. Let $g : X \to \{0, 1\}$ be an arbitrary function and let $p_1 = \mathbf{E}_{x \sim \mu}[f(x)]$. Let $D = (\mu, f)$ and $D' = (\mu, f')$. We relate the quantities false_+ and false_- of g with respect to the D and D' – simple calculations show that

$$\text{false}_+\big(g, D'\big) = \text{false}_+(g, D) + \left(\frac{1}{2} - \frac{\epsilon}{4}\right)\big(p_1 - \text{false}_-(g, D)\big), \tag{1}$$

$$\text{false}_-\big(g, D'\big) = \left(\frac{1}{2} + \frac{\epsilon}{4}\right)\text{false}_-(g, D), \tag{2}$$

$$\text{err}\big(g, D'\big) = \text{false}_+(g, D) + \left(\frac{1}{2} - \frac{\epsilon}{4}\right)p_1 + \frac{\epsilon}{2}\text{false}_-(g, D). \tag{3}$$

Let $c \in C$ be such that $\text{false}_+(c, D) = 0$ and $\text{false}_-(c, D) = \text{opt}_+ = \min_{c' \in C^+} \text{false}_-(c', D)$. If h is the output of $\mathcal{A}^{\mathcal{O}(f')}(\frac{\epsilon^2}{2}, \delta)$, it satisfies the following with probability at least $1 - \delta$,

$$\text{err}\big(h, D'\big) \leqslant \text{err}\big(c, D'\big) + \frac{\epsilon^2}{2}. \tag{4}$$

Substituting identity (3) in (4), once for c and once for h we get

$$\begin{aligned} \text{false}_+(h, D) &\leqslant \frac{\epsilon}{2}\big(\text{opt}_+ - \text{false}_-(h, D)\big) + \frac{\epsilon^2}{2} \\ &\leqslant \epsilon. \end{aligned} \tag{5}$$

Dropping the non-negative term $\text{false}_+(h, D)$ from (5) and rearranging we get

$$\begin{aligned} &\frac{\epsilon}{2}\text{false}_-\big(h, D'\big) \leqslant \frac{\epsilon}{2}\text{opt}_+ + \frac{\epsilon^2}{2}, \\ &\text{false}_-\big(h, D'\big) \leqslant \text{opt}_+ + \epsilon. \qquad \square \end{aligned} \tag{6}$$

The algorithm for learning disjunctions of concepts requires as an intermediate step, learning over subsets of the instance space. The key idea here is that boosting is easy in the case of positive-reliable learning. Since the classifiers we obtain at each step have (almost) no false positives, in the next step we only need to consider the subset of examples that have so far been classified negative. Running the positive-reliable learning algorithm again on this subset of examples will give a useful hypothesis to add to the disjunction.

We show that in the agnostic setting, learning on a subset of the instance space is only as hard as learning over the entire instance space. The simple reduction we present here does not seem feasible in the case of PAC learning. Let $S \subseteq X$ be a subset, a distribution μ over X induces a conditional distribution $\mu|_S$ over S as follows – for any $T \subseteq X$, $\Pr_{\mu|_S}[T] = \Pr_\mu[T \cap S] / \Pr_\mu[S]$. We assume access to the *indicator function* for the set S, i.e. $I_S : X \to \{0, 1\}$ such that $I_S(x) = 1$ if $x \in S$

and $I_S(x) = 0$ otherwise. If $\Pr_\mu[S]$ is not negligible, it is possible to agnostically learn under conditional distribution $\mu|_S$ using a black-box agnostic learner that has guarantees under μ.

Lemma 3. *Suppose algorithm $\mathcal{A}^{\mathcal{O}(f)}(\epsilon, \delta)$ efficiently agnostically learns C under distribution μ over X in time $T(\epsilon, \delta)$. Let $S \subseteq X$ such that, $\Pr_\mu[S] \geqslant \gamma$. Then, algorithm $\mathcal{A}^{\mathcal{O}(f')}(\epsilon\gamma, \delta)$, where $f' = fI_S + (1 - I_S)/2$, efficiently agnostically learns C under distribution $\mu|_S$ in time $T(\epsilon\gamma, \delta)$.*

Proof. We first relate the errors of an arbitrary boolean hypothesis g with respect to two target functions f and $f' = fI_S + (1 - I_S)/2$. In words, f' is f on S and random outside S. Therefore, the error of any hypothesis g would be $1/2$ outside of S and its error under $\mu|_S$ inside S. Let $D = (\mu, f)$, $D' = (\mu, f')$ and $D|_S = (\mu|_S, f)$. Formally, for any function $g : X \to \{0, 1\}$ we have

$$\begin{aligned}
\operatorname{err}\big(g, D'\big) &= \mathop{\mathbf{E}}_{x\sim\mu}\big[g(x)\big(1 - f'(x)\big) + \big(1 - g(x)\big)f'(x)\big] \\
&= \mathop{\mathbf{E}}_{x\sim\mu}\big[\big(g(x)\big(1 - f(x)\big) + \big(1 - g(x)\big)f(x)\big)I_S\big] + \mathop{\mathbf{E}}_{x\sim\mu}\big[(1 - I_S)/2\big] \\
&= \Pr_\mu[S]\operatorname{err}(g, D|_S) + \Big(1 - \Pr_\mu[S]\Big)/2. \qquad (7)
\end{aligned}$$

Let $c \in C$ be such that $\operatorname{err}(c, D|_S) = \min_{c' \in C}\operatorname{err}(c', D|_S) = \operatorname{opt}$. Using (7) we can conclude that c is also optimal under joint distribution D'. Algorithm $\mathcal{A}^{\mathcal{O}(f')}(\epsilon\gamma, \delta)$ outputs hypothesis h which satisfies with probability at least $1 - \delta$, $\operatorname{err}(h, D') \leqslant \operatorname{err}(c, D') + \epsilon\gamma$. Using (7) and since $\Pr_\mu[S] \geqslant \gamma$ we immediately get $\operatorname{err}(h, D|_S) \leqslant \operatorname{opt} + \epsilon$. □

We define algorithm CPRL (conditional positive-reliable learner) as doing the following – the inputs to the algorithm are:

- ϵ – the accuracy parameter,
- δ – the confidence parameter,
- γ – lower bound on the probability mass of S,
- $\mathcal{O}(f)$ – oracle to access the target function,
- I_S – the indicator function for subset S.

Algorithm CPRL initially defines $f'' = (1/2 + \epsilon/4)(fI_S + (1 - I_S)/2)$. By Lemma 1 we can simulate $\mathcal{O}(f'')$ given access to $\mathcal{O}(f)$ and I_S. Algorithm CPRL runs $\mathcal{A}^{\mathcal{O}(f'')}(\epsilon^2\gamma^2/2, \delta)$ and returns its output. Algorithm CPRL thus returns a hypothesis with performance close to the best positive-reliable classifier on the subset S (using Lemmas 2 and 3). We state the properties of algorithm CPRL as Lemma 4. The proof is a direct application of Lemmas 2 and 3.

Lemma 4. *Suppose algorithm $\mathcal{A}^{\mathcal{O}(f)}(\epsilon, \delta)$ agnostically learns C under distribution μ (over X) in time $T(\epsilon, \delta)$. For a subset $S \subseteq X$ with $\Pr_\mu[S] \geqslant \gamma$, any target function f (inducing $D|_S = (\mu|_S, f)$) algorithm* CPRL *(which uses $\mathcal{A}$ as a black-box) with probability at least $1 - \delta$ returns a hypothesis h such that* $\operatorname{false}_+(h, D|_S) \leqslant \epsilon$ *and*

$$\operatorname{false}_-(h, D|_S) \leqslant \min_{c \in C^+}\operatorname{false}_-(c, D|_S) + \epsilon.$$

Algorithm CPRL *has access to oracle $\mathcal{O}(f)$ and black-box access to I_S. The running time of algorithm* CPRL *is $T(\epsilon^2\gamma^2/2, \delta)$.*

To prove the result on learning disjunctions, we will use two simple lemmas. Lemma 5 states that if a hypothesis h has a rate of false negatives not much more (within ϵ) than that of a concept c that predicts no false positives, then h must predict 1 at least as often (within ϵ) as c. Lemma 6 states that if h is a hypothesis with low rate of false positives and if it predicts 1 at least as often (within $\epsilon/2$) as a concept c which makes no false positive errors, the rate of false negatives of h must not be much more (within ϵ) than that of c.

Lemma 5. *Let μ be a distribution over X, $f : X \to [0, 1]$ an arbitrary target function (with $D = (\mu, f)$) and let c, h be such that* $\operatorname{false}_+(c, D) = 0$ *and* $\operatorname{false}_-(h, D) \leqslant \operatorname{false}_-(c, D) + \epsilon$, *then* $\Pr_\mu[h(x) = 1] \geqslant \Pr_\mu[c(x) = 1] - \epsilon$.

Proof.

$$\begin{aligned}
\Pr_\mu\big[h(x) = 1\big] &= 1 - \Pr_\mu\big[h(x) = 0\big] \\
&= 1 - \Pr_{(x,y)\sim D}\big[h(x) = 0 \wedge y = 0\big] - \Pr_{(x,y)\sim D}\big[h(x) = 0 \wedge y = 1\big] \\
&\geqslant 1 - \Pr_{(x,y)\sim D}[y = 0] - \operatorname{false}_-(h, D).
\end{aligned}$$

Algorithm: DISJUNCTION-LEARNER

input: ϵ, δ, M, $\mathcal{O}(f)$, CPRL
set $H_0 := 0$;
for $i = 1$ to M {
  set $m := \frac{2}{\epsilon^2}\log\frac{2M}{\delta}$;
  draw $Z = \langle (x_1, y_1), \ldots, (x_m, y_m)\rangle$ from (D, f)
  set $\hat{p}_i := \sum_{j=1}^{m}(1 - H_{i-1}(x_j))$;
  if $\hat{p}_i \geqslant \frac{\epsilon}{2}$ {
    $h_i := \mathsf{CPRL}(\frac{\epsilon}{4M}, \frac{\delta}{2M}, \frac{2}{3}\hat{p}_i, \mathcal{O}(f), 1 - H_{i-1})$;
  } **else** {
    $h_i := 0$;
  }
  $H_i := H_{i-1} \vee h_i$;
}
output H_M

Fig. 2. Algorithm DISJUNCTION-LEARNER.

Since $\text{false}_+(c, D) = 0$, $\Pr_{(x,y)\sim D}[c(x) = 0 \wedge y = 0] = \Pr_{(x,y)\sim D}[y = 0]$. Thus,

$$\begin{aligned}
&\geqslant 1 - \Pr_{(x,y)\sim D}\big[c(x) = 0 \wedge y = 0\big] - \text{false}_-(c, D) - \epsilon \\
&\geqslant 1 - \Pr_{\mu}\big[c(x) = 0\big] - \epsilon \\
&\geqslant \Pr_{\mu}\big[c(x) = 1\big] - \epsilon. \qquad \square
\end{aligned}$$

Lemma 6. *Let μ be a distribution over X, $f : X \to [0, 1]$ an arbitrary target function (with $D = (\mu, f)$) and let c, h be such that $\text{false}_+(c, D) = 0$, $\text{false}_+(h, D) \leqslant \epsilon/2$ and $\Pr_\mu[h(x) = 1] \geqslant \Pr_\mu[c(x) = 1] - \epsilon/2$, then $\text{false}_-(h, D) \leqslant \text{false}_-(c, D) + \epsilon$.*

Proof.

$$\begin{aligned}
\text{false}_-(h, D) &= \Pr_{(x,y)\sim D}\big[h(x) = 0 \wedge y = 1\big] \\
&= \Pr_{\mu}\big[h(x) = 0\big] - \Pr_{(x,y)\sim D}\big[h(x) = 0 \wedge y = 0\big] \\
&\leqslant \Pr_{D}\big[c(x) = 0\big] + \epsilon/2 - \Pr_{D}[y = 0] + \Pr_{(x,y)\sim D}\big[h(x) = 1 \wedge y = 0\big].
\end{aligned}$$

Since $\text{false}_+(c) = 0$, $\Pr_{(x,y)\sim D}[c(x) = 0 \wedge y = 0] = \Pr_{(x,y)\sim D}[y = 0]$. Thus,

$$\begin{aligned}
\text{false}_-(h, D) &\leqslant \Pr_{(x,y)\sim D}\big[c(x) = 0 \wedge y = 1\big] + \epsilon/2 + \text{false}_+(h, D) \\
&= \text{false}_-(c, D) + \epsilon. \qquad \square
\end{aligned}$$

Let $\mathsf{OR}_s(C) = \{c_1 \vee \cdots \vee c_s \mid c_j \in C,\ j \leqslant s\}$ be the class of size-s disjunctions of concepts from C. Below, we state and prove Theorem 2 which is a formal restatement of Theorem 1 which states that if C is agnostically learnable under distribution μ, then $\mathsf{OR}_s(C)$ is positive-reliably learnable under distribution μ. Algorithm DISJUNCTION-LEARNER (Fig. 2) is a boosting-like algorithm which positive-reliably learns the disjunctions of concepts from C using a black-box agnostic learner for C. The proof is a simple covering argument using Lemmas 5 and 6. The key idea in the proof is to show that by adding a new (almost) positive-reliable classifier at each step, we cover as many positive examples (i.e. predict positive almost as often), as the best positive-reliable disjunction, at the same time making almost no false positive errors. As long as the true disjunction predicts positive more often than our current hypothesis, we can always find a new (almost) positive-reliable classifier that can be added to our current hypothesis driving up the measure of our positive predictions. Since when we terminate our hypothesis has very few false positive errors and predicts positive (almost) as often the true disjunction, our final hypothesis cannot have false negative error rate much more than the best positive-reliable disjunction (cf. Lemma 6). The reader is referred to Fig. 2 for some notation used in the proof of Theorem 2.

Theorem 2. *Let C be a concept class that is agnostically learnable (in polynomial time) under μ and $f : X \to [0, 1]$ be an arbitrary target function ($D = (\mu, f)$). Algorithm* DISJUNCTION-LEARNER *(see Fig. 2) run with parameters ϵ, δ, $M = s\log(4/\epsilon)$ and access to oracle $\mathcal{O}(f)$ and black-box access to* CPRL*, with probability at least $1 - \delta$ outputs a hypothesis H_M, such that $\text{false}_+(H_M, D) \leqslant \epsilon$ and*

$$\text{false}_-(H_M, D) \leqslant \min_{\psi \in \mathsf{OR}_s(C)^+} \text{false}_-(\psi, D) + \epsilon.$$

The running time is polynomial in s, $\frac{1}{\epsilon}$, $\frac{1}{\delta}$.

Proof. For definitions of H_i and $\hat{p}_i$ refer to Fig. 2. Let $\varphi \in \mathsf{OR}_s(C)^+$ satisfy $\text{false}_+(\varphi, D) = 0$ and $\text{false}_-(\varphi, D) = \text{opt}_+ = \min_{\psi \in \mathsf{OR}_s(C)^+} \text{false}_-(\psi, D)$. Suppose $\varphi = c_1 \vee \cdots \vee c_s$, where $c_j \in C$. Let $S_i = \{x \in X \mid H_{i-1}(x) = 0\}$, and hence $1 - H_{i-1}$ is an indicator function for S_i. Let $p_i = \Pr_\mu[S_i]$ and $p_+ = \Pr_\mu[\varphi(x) = 1]$. Our goal is to show that $\Pr_\mu[H_M(x) = 1] \geqslant p_+ - \epsilon/2$ and $\text{false}_+(H_M, D) \leqslant \epsilon/2$ and then using Lemma 6 we are done.

In the ith iteration, we compute $\hat{p}_i$ to estimate p_i, using a sample of size m. Using Hoeffding's bound, with probability at least $1 - (\delta/2M)$, $|\hat{p}_i - p_i| \leqslant (\epsilon/2)$. Let us suppose this holds for all M iterations, allowing our algorithm a failure probability of $\delta/2$ so far. In this case, if $p_i \geqslant \epsilon$, $\hat{p}_i \geqslant (\epsilon/2)$ and $\hat{p}_i \leqslant (3p_i/2)$ (thus the call to CPRL in the algorithm is valid). Note that if $p_i < \epsilon$, then $\Pr_\mu[H_{i-1}(x) = 1] \geqslant 1 - \epsilon$, and hence $\text{false}_-(H_{i-1}, D) < \epsilon$; for all iterations it is easy to see that $\text{false}_+(H_{i-1}, D) \leqslant \epsilon$ (since $\text{false}_+(H_{i-1}, D) \leqslant \sum_{j=1}^{i-1} \text{false}_+(h_j, D)$) and hence in this case we are done. Let $\mu|_{S_i}$ denote the conditional distribution given S_i and $D|_{S_i} = (\mu|_{S_i}, f)$. In the ith iteration the call to CPRL returns a hypothesis h_i such that $\text{false}_+(h_i, D|_{S_i}) \leqslant \frac{\epsilon}{4M}$ and $\text{false}_-(h_i, D|_{S_i}) \leqslant \text{opt}_i + \frac{\epsilon}{4M}$ where $\text{opt}_i = \min_{c \in C^+} \text{false}_-(c, D|_{S_i})$. Note that for $j = 1, \ldots, s$, $\text{false}_+(c_j, D|_{S_i}) = 0$, and hence $\text{false}_-(c_j, D|_{S_i}) \geqslant \text{opt}_i$ for all i. Thus we get

$$\text{false}_-(h_i, D|_{S_i}) \leqslant \text{false}_-(c_j, D|_{S_i}) + \frac{\epsilon}{4M}$$

and hence using Lemma 5,

$$\Pr_{\mu|_{S_i}}\big[h_i(x) = 1\big] \geqslant \Pr_{\mu|_{S_i}}\big[c_j(x) = 1\big] - \frac{\epsilon}{4M},$$

$$\Pr_{\mu|_{S_i}}\big[h_i(x) = 1\big] \geqslant \frac{1}{s} \Pr_{\mu|_{S_i}}\big[\varphi(x) = 1\big] - \frac{\epsilon}{4M}.$$

We define the quantity $d_i = p_+ - \Pr_\mu[H_i(x) = 1]$ and show by induction that $d_i \leqslant p_+(1 - \frac{1}{s})^i + \frac{\epsilon}{4M} \sum_{j=0}^{i-1} (1 - \frac{1}{s})^j$. To check the base step see that $d_1 = p_+ - \Pr_\mu[h_1(x) = 1] \leqslant p_+ - \frac{p_+}{s} + \frac{\epsilon}{4M}$. For the induction step we have

$$\begin{aligned}
d_{i+1} &= p_+ - \Pr_\mu\big[H_{i+1}(x) = 1\big] \\
&= p_+ - \Pr_\mu\big[H_i(x) = 1\big] - \Pr_\mu\big[H_i(x) = 0 \wedge h_{i+1}(x) = 1\big] \\
&= d_i - \Pr_{\mu|_{S_{i+1}}}\big[h_{i+1}(x) = 1\big] \Pr_\mu\big[H_i(x) = 0\big] \\
&\leqslant d_i - \frac{1}{s} \Pr_{\mu|_{S_{i+1}}}\big[\varphi(x) = 1\big] \Pr_\mu\big[H_i(x) = 0\big] + \frac{\epsilon}{4M} \\
&\leqslant d_i - \frac{1}{s} \Pr_\mu\big[\varphi(x) = 1 \wedge H_i(x) = 0\big] + \frac{\epsilon}{4M} \\
&\leqslant d_i - \frac{1}{s}\Big(\Pr_\mu\big[\varphi(x) = 1\big] - \Pr_\mu\big[H_i(x) = 1\big]\Big) + \frac{\epsilon}{4M} \\
&\leqslant d_i\left(1 - \frac{1}{s}\right) + \frac{\epsilon}{4M} \\
&\leqslant p_+\left(1 - \frac{1}{s}\right)^{i+1} + \frac{\epsilon}{4M} \sum_{j=0}^{i} \left(1 - \frac{1}{s}\right)^j.
\end{aligned}$$

When $M = s \log \frac{4}{\epsilon}$, $d_M \leqslant \epsilon/2$, and it is easily seen that $\text{false}_+(H_M, D) \leqslant \epsilon/2$ and hence using Lemma 6, $\text{false}_-(H_M, D) \leqslant \text{opt}_+ + \epsilon$. The probability of failure of some call to CPRL is at most $M \frac{\delta}{2M} = \delta/2$, which combined with probability of failure caused by incorrect estimation of some $\hat{p}_i$ gives total failure probability at most δ. □

Theorem 2 combined with Lemma 7 below (which is Theorem 18 from [8]) proves Corollary 1.

Lemma 7 *(Gopalan, Kalai and Klivans (2008)). The class of polynomial-size decision trees can be agnostically learned (using queries) to accuracy ϵ in time* $\text{poly}(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ *with respect to the uniform distribution.*

4. Full reliability

In this section we define the notion of full reliability. The goal is to obtain a hypothesis that has low error rate in terms of false positives and negatives both. In the noisy (agnostic) setting, this is not possible unless we allow the hypothesis to refrain from making a prediction. For this reason we use the notion of *partial classifiers* which predict a value from the set $\{0, 1, ?\}$, where prediction of "?" is treated as uncertainty of the classifier. Recall that our instance space is $\langle X_n \rangle_{n \geqslant 1}$

with distributions $\langle \mu_n \rangle_{n \geqslant 1}$ and for each n, the target is an unknown function $f_n : X \to [0,1]$ inducing a joint distribution $D_n = (\mu_n, f_n)$ over $X_n \times \{0,1\}$. Formally, a partial classifier is $c : X_n \to \{0,1,?\}$. Let $\mathrm{I}(\mathcal{E})$ denote the indicator function for event $\mathcal{E}$. Then false_+, false_- and error for a partial classifier can be defined as

$$\mathrm{false}_+(c, D_n) = \mathop{\mathbf{E}}_{x \sim \mu_n}\big[\mathrm{I}\big(c(x) = 1\big)\big(1 - f_n(x)\big)\big],$$

$$\mathrm{false}_-(c, D_n) = \mathop{\mathbf{E}}_{x \sim \mu_n}\big[\mathrm{I}\big(c(x) = 0\big) f_n(x)\big],$$

$$\mathrm{err}(c, D_n) = \mathop{\mathbf{E}}_{x \sim D_n}\big[\mathrm{I}\big(c(x) = 0\big) f_n(x) + \mathrm{I}\big(c(x) = 1\big)\big(1 - f_n(x)\big)\big].$$

We define the *uncertainty* of a partial classifier c to be

$$?(c, D_n) = \mathop{\mathbf{E}}_{x \sim D_n}\big[\mathrm{I}\big(c(x) = ?\big)\big].$$

Finally we define the *accuracy* of a partial classifier c to be

$$\mathrm{acc}(c, D_n) = \mathop{\mathbf{E}}_{x \sim D_n}\big[\mathrm{I}\big(c(x) = 0\big)\big(1 - f_n(x)\big) + \mathrm{I}\big(c(x) = 1\big) f_n(x)\big].$$

Accuracy can be interpreted as the probability that c makes a correct prediction. For a partial classifier c, we have $\mathrm{err}(c) + ?(c) + \mathrm{acc}(c) = 1$.

Suppose that $\langle C_n \rangle_{n \geqslant 1}$ is a sequence of concept classes; recall that we defined $C_n^+ = \{c \in C_n \mid \mathrm{false}_+(c) = 0\}$, the subset of positive-reliable classifiers from C_n. Similarly we define $C_n^- = \{c \in C_n \mid \mathrm{false}_-(c) = 0\}$, the subset of negative-reliable classifiers. We can now define the class of partial classifiers derived from C_n as $\mathsf{PC}(C_n) = \{(c_+, c_-) \mid c_+ \in C_n^+,\ c_- \in C_n^-\}$. The definition ensures that for a partial classifier $c = (c_+, c_-) \in \mathsf{PC}(C_n)$, $x \in X_n$ drawn from μ_n satisfies $c_-(x) \geqslant c_+(x)$[4] with probability 1. For any x, $c(x) = 1$, when both $c_+(x) = 1$ and $c_-(x) = 1$, $c(x) = 0$, when both $c_+(x) = 0$ and $c_-(x) = 0$ and $c(x) = ?$ otherwise. $\mathsf{PC}(C_n)$ is a set of partial classifiers, derived from C_n, which make no errors, i.e. the class of fully-reliable classifiers. We define fully-reliable learning as:

Fully-reliable learning. Algorithm $\mathcal{A}$ *efficiently fully-reliably learns* a sequence of concept classes $\langle C_n \rangle_{n \geqslant 1}$ under distributions $\langle \mu_n \rangle_{n \geqslant 1}$, if there exists a polynomial $p(n, 1/\epsilon, 1/\delta)$ such that, for every $n \geqslant 1$, $\epsilon, \delta > 0$ and every $f_n : X_n \to [0,1]$ (inducing $D_n = (\mu_n, f_n)$), algorithm $\mathcal{A}^{\mathcal{O}(f_n)}(\epsilon, \delta)$, with probability at least $1 - \delta$, outputs a hypothesis h such that $\mathrm{err}(h, D_n) \leqslant \epsilon$ and

$$\mathrm{acc}(h, D_n) \geqslant \max_{c \in \mathsf{PC}(C_n)} \mathrm{acc}(c, D_n) - \epsilon.$$

The time complexity of both $\mathcal{A}$ and h is bounded by $p(n, 1/\epsilon, 1/\delta)$.

In Section 3 we defined algorithm PRL for positive-reliable learning using a black-box agnostic learner for $\langle C \rangle_{n \geqslant 1}$ (cf. Lemma 2). A symmetric algorithm NRL (negative-reliable learner) can be defined, again using a black-box agnostic learning, for negative-reliable learning. Our main result of this section is showing that agnostic learning implies fully-reliable learning.

Theorem 3. *If C is efficiently agnostically learnable under distribution μ, it is efficiently fully-reliably learnable under distribution μ.*

Proof. A simple algorithm to fully-reliably learn C is the following: Let $h_+ = \mathsf{PRL}^{\mathcal{O}(f)}(\frac{\epsilon}{4}, \frac{\delta}{2})$, $h_- = \mathsf{NRL}^{\mathcal{O}(f)}(\frac{\epsilon}{4}, \frac{\delta}{2})$. Define $h : X \to \{0,1,?\}$ as

$$h(x) = \begin{cases} 1 & \text{if } h_+(x) = h_-(x) = 1, \\ 0 & \text{if } h_+(x) = h_-(x) = 0, \\ ? & \text{otherwise.} \end{cases}$$

We claim that hypothesis h is close to the best fully-reliable partial classifier.

For $D = (\mu, f)$, let $c = (c_+, c_-) \in \mathsf{PC}(C)$ be such that $\mathrm{err}(c, D) = 0$ and $\mathrm{acc}(c, D) = \max_{c' \in \mathsf{PC}(C)} \mathrm{acc}(c', D)$. Using properties of PRL, we know that the following hold:

$$\mathrm{false}_+(h_+, D) \leqslant \frac{\epsilon}{4},$$

$$\mathrm{false}_-(h_+, D) \leqslant \mathrm{false}_-(c_+, D) + \frac{\epsilon}{4}.$$

[4] If $c_+(x) = 1$, then $f(x) = 1$ since c_+ is positive-reliable and hence $c_-(x) = 1$ since c_- is negative-reliable. When $c_+(x) = 0$, in any case $c_-(x) \geqslant c_+(x)$ holds.

Algorithm: SANDWICH-LEARNER

inputs: m, T, d

1. Draw m labeled examples $Z^t = \langle (x^{t_1}, y^{t_1}), \ldots, (x^{t_m}, y^{t_m}) \rangle \in (\mathcal{B}_n \times \{0,1\})^m$ and set up the ℓ_1 polynomial regression problem (8)–(11).
2. Solve the regression problem, get polynomials $p^t(x)$, $q^t(x)$ and record z^t to be the value of the objective function.
3. Repeat the above two steps T times and take $p = p^s$, $q = q^s$ to be the polynomials where z^s was the least.
4. Output a randomized partial hypothesis $h : \mathcal{B}_n \times \{0,1,?\} \to [0,1]$:
   $h(x,1) = \text{crop}(\min(p(x), q(x)))$
   $h(x,0) = \text{crop}(\min(1-p(x), 1-q(x)))$
   $h(x,?) = 1 - h(x,1) - h(x,0)$

Fig. 3. Algorithm SANDWICH-LEARNER.

Also, the hypothesis h_- output by NRL satisfies

$$\text{false}_-(h_-, D) \leqslant \frac{\epsilon}{4},$$
$$\text{false}_+(h_-, D) \leqslant \text{false}_+(c_-, D) + \frac{\epsilon}{4}.$$

Then $\text{err}(h, D) \leqslant \text{false}_+(h_+, D) + \text{false}_-(h_-, D) \leqslant \frac{\epsilon}{2}$.

Note that when $h(x) = ?$, $h_+(x) \neq h_-(x)$ and hence exactly one of them makes an error. Thus we have $\text{err}(h, D) + ?(h, D) \leqslant \text{err}(h_+, D) + \text{err}(h_-, D)$. Therefore, we also have $1 - \text{acc}(h, D) \leqslant \text{err}(h_+, D) + \text{err}(h_-, D)$. We know that

$$\begin{aligned}\text{err}(h_+, D) &= \text{false}_+(h_+, D) + \text{false}_-(h_+, D) \\ &\leqslant \text{false}_-(c_+, D) + \frac{\epsilon}{2}.\end{aligned}$$

Similarly $\text{err}(h_-, D) \leqslant \text{false}_+(c_-, D) + \epsilon/2$. Finally, we check that $\text{false}_-(c_+, D) + \text{false}_+(c_-, D) = 1 - \text{acc}(c, D)$, hence $\text{acc}(h, D) \geqslant \text{acc}(c, D) - \epsilon$. □

5. Reliable learning with tolerance

In this section, we consider our final generalization where we allow tolerance rates τ_+ and τ_- for false positives and false negatives respectively. As in the case of fully-reliable learning we need to consider the class of partial classifiers, i.e. those predicting from $\{0, 1, ?\}$. Given a sequence of concept classes $\langle C_n \rangle_{n \geqslant 1}$, for each C_n we define the class of sandwich classifiers as $\text{SC}(C_n) = \{(c_+, c_-) \mid c_+ \leqslant c_-\}$. For a sandwich classifier $c = (c_+, c_-)$, given x, $c(x) = 1$ when both $c_+(x) = 1$ and $c_-(x) = 1$, $c(x) = 0$ when both $c_+(x) = 0$ and $c_-(x) = 0$ and "?" otherwise.

Sandwich classifiers are a generalization of the class of partial classifiers $\text{PC}(C_n)$ that we defined in the previous section. In particular, the class $\text{PC}(C_n)$ is the set of all sandwich classifiers with zero error. We call a sandwich classifier $c \in \text{SC}(C_n)$, (τ_+, τ_-)-*tolerant reliable* if $\text{false}_+(c, D_n) \leqslant \tau_+$ and $\text{false}_-(c, D_n) \leqslant \tau_-$. Given acceptable tolerance rates of τ_+, τ_- define opt as

$$\text{opt}(\tau_+, \tau_-) = \max_{c \in \text{SC}(C_n)} \text{acc}(c);$$
$$\text{subject to:} \quad \{\text{false}_+(c) \leqslant \tau_+;\ \text{false}_-(c) \leqslant \tau_-\}.$$

We show that algorithm SANDWICH-LEARNER (Fig. 3) efficiently learns the class of halfspace sandwiches, over the unit ball $\mathcal{B}_n$, under the uniform distribution. With some modifications, this algorithm can also be made to learn under log-concave distributions. A halfspace sandwich over $\mathcal{B}_n$ can be visualized as two slices, one labeled 1, the other 0 and the remaining ball labeled "?" (see Fig. 1(c)). Our algorithm uses techniques and analysis of the agnostic halfspace algorithm due to Kalai, Klivans, Mansour and Servedio [12]. Algorithm SANDWICH-LEARNER draws a sample of size m in each iteration and sets up the following ℓ_1 regression problem (8)–(11) over polynomials p and q:

$$\min_{\substack{\deg(p) \leqslant d \\ \deg(q) \leqslant d}} \frac{1}{m} \left(\sum_{i:\ y^i = 1} |1 - p(x^i)| + \sum_{i:\ y^i = 0} |q(x^i)| \right) \tag{8}$$

subject to

$$\frac{1}{m}\sum_{i:\ y^i=0}\left|p\left(x^i\right)\right| \leqslant \tau_+ + \frac{\epsilon}{2}, \tag{9}$$

$$\frac{1}{m}\sum_{i:\ y^i=1}\left|1-q\left(x^i\right)\right| \leqslant \tau_- + \frac{\epsilon}{2}, \tag{10}$$

$$\frac{1}{m}\sum_{i}\max\left(p\left(x^i\right)-q\left(x^i\right),0\right) \leqslant \frac{\epsilon}{8}. \tag{11}$$

This problem can be solved efficiently using linear programming (cf. Appendix A of [12]). Thus, the running time of the algorithm is polynomial in m, T and d. The function $\mathrm{crop}:\mathbb{R}\to[0,1]$ used in step 4 of the algorithm is defined as

$$\mathrm{crop}(z)=\begin{cases} z & \text{for } z\in[0,1],\\ 0 & \text{for } z<0,\\ 1 & \text{for } z>1.\end{cases}$$

The output of the algorithm is a randomized partial classifier. We define a *randomized partial classifier* as a function, $c: X\times\{0,1,?\}\to[0,1]$, so that $c(x,0)+c(x,1)+c(x,?)=1$, and $c(x,y)$ is interpreted as the probability that the output for x is y. We define error and empirical error of such classifiers below. Here Z is a sample of size m drawn from distribution $D=(\mu,f)$.

$$\mathrm{err}(c,D)=\mathop{\mathbf{E}}_{(x,y)\sim D}\left[c(x,1-y)\right],$$

$$\widehat{\mathrm{err}}(c,Z)=\frac{1}{m}\sum_{i=1}^{m}c\left(x^i,1-y^i\right).$$

Other quantities (such as acc and uncertainty ("?")) are defined similarly. Theorem 4 below shows that for any constant ϵ, the class of halfspace sandwiches over the unit ball $\mathcal{B}_n$ is efficiently (τ_+,τ_-)-tolerant reliably learnable. The running time of the algorithm is exponential in $1/\epsilon$.

Theorem 4. *Let $\mathcal{U}$ be the uniform distribution on the unit ball $\mathcal{B}_n$ and let $f:\mathcal{B}_n\to[0,1]$ be an arbitrary target function. There exists a polynomial P such that, for any $\epsilon,\delta>0$, $n\geqslant 1$, $\tau_+,\tau_-\geqslant 0$, algorithm* Sandwich-Learner *with parameters $m=P(n^d/(\epsilon\delta))$, $T=\log(2/\delta)$, $d=O(1/\epsilon^4)$, with probability at least $1-\delta$, returns a randomized hypothesis h which satisfies* $\mathrm{false}_+(h)\leqslant\tau_++\epsilon$, $\mathrm{false}_-(h)\leqslant\tau_-+\epsilon$ *and* $\mathrm{acc}(h)\geqslant\mathrm{opt}(\tau_+,\tau_-)-\epsilon$, *where* opt *is with respect to the class of halfspace sandwich classifiers.*

To prove Theorem 4, we need Lemma 8 as an intermediate step. Here $\mathcal{B}_n$ is the unit ball in n dimensions.

Lemma 8. *Let μ be a distribution over $\mathcal{B}_n$ and $f:\mathcal{B}_n\to[0,1]$ be an arbitrary function (with $D=(\mu,f)$). Let τ_+, τ_- be the required tolerance parameters. Suppose $c=(c_+,c_-)$ is a halfspace sandwich classifier such that* $\mathrm{false}_+(c)\leqslant\tau_+$, $\mathrm{false}_-(c)\leqslant\tau_-$ *and* $\mathrm{acc}(c)=\mathrm{opt}(\tau_+,\tau_-)$. *If p_+, p_- are degree d polynomials such that $\mathbf{E}[|c_+(x)-p_+(x)|]\leqslant\frac{\epsilon}{128}$ and $\mathbf{E}[|c_-(x)-p_-(x)|]\leqslant\frac{\epsilon}{128}$, and if $m=\frac{128}{\epsilon^2}$, then with probability at least $\frac{1}{2}$, $p=p_+$, $q=p_-$ is a feasible solution to the ℓ_1* Polynomial Regression Problem (8)–(11) *and the value of the objective function is at most* $\mathrm{false}_-(c_+)+\mathrm{false}_+(c_-)+\frac{\epsilon}{2}$.

Proof. Using Hoeffding's bound when $m=\frac{128}{\epsilon^2}$,

$$\Pr\left[\widehat{\mathrm{false}}_+(c_+)\geqslant\mathrm{false}_+(c_+)+\frac{\epsilon}{8}\right]\leqslant\frac{1}{16} \tag{12}$$

where $\widehat{\mathrm{false}}_+(c_+)$ is the empirical estimate of $\mathrm{false}_+(c_+)$ using a sample of size m. Also by Markov's inequality,

$$\Pr\left[\frac{1}{m}\sum_i\left|c_+\left(x^i\right)-p_+\left(x^i\right)\right|\geqslant\frac{\epsilon}{8}\right]\leqslant\Pr\left[\frac{1}{m}\sum_i\left|c_+\left(x^i\right)-p_+\left(x^i\right)\right|\geqslant 16\,\mathbf{E}\left[\left|c_+(x)-p_+(x)\right|\right]\right]\leqslant\frac{1}{16}. \tag{13}$$

And hence with probability at least 7/8,

$$\begin{aligned}\frac{1}{m}\sum_{i:\ y^i=0}\left|p_+\left(x^i\right)\right| &\leqslant \frac{1}{m}\sum_{i:\ y^i=0}\left(\left|c_+\left(x^i\right)\right|+\left|c_+\left(x^i\right)-p_+\left(x^i\right)\right|\right)\\ &\leqslant \mathrm{false}_+(c_+)+\epsilon/8+\epsilon/8\\ &\leqslant \tau_++\epsilon/4\end{aligned} \tag{14}$$

and hence $p=p_+$ satisfies constraint (9) in the regression problem.

Similarly one can show that with probability at least 7/8, $q = p_-$ satisfies constraint (10) of the regression problem. Below we use the fact that $c_-(x) \geqslant c_+(x)$ for all x (according to the definition of $\mathsf{SC}(C)$). Consider the following:

$$\begin{aligned}\mathbf{E}\big[\max\big(p_+(x) - p_-(x), 0\big)\big] &\leqslant \mathbf{E}\big[\big|p_+(x) - p_-(x) + c_-(x) - c_+(x)\big|\big] \\ &\leqslant \mathbf{E}\big[\big|p_+(x) - c_+(x)\big|\big] + \mathbf{E}\big[\big|p_-(x) - c_-(x)\big|\big] \leqslant \epsilon/64.\end{aligned}$$

Then by Markov's inequality,

$$\Pr\left[\frac{1}{m}\sum_i \max\big(p_+\big(x^i\big) - p_-\big(x^i\big), 0\big) \geqslant \frac{\epsilon}{8}\right] \leqslant \frac{1}{8}. \tag{15}$$

By union bound with probability at least 5/8, with $p = p_+$, $q = p_-$ all constraints of the regression problem are satisfied. Let us assume we are in the event where all of (12)–(15) and the corresponding statements in the case of false negatives are true. Using Hoeffding's bound, $\Pr[\widehat{\text{false}}_-(c_+) \geqslant \text{false}_-(c_+) + \epsilon/8] \leqslant 1/16$ and $\Pr[\widehat{\text{false}}_+(c_-) \geqslant \text{false}_+(c_-) + \epsilon/8] \leqslant 1/16$. We allow ourselves a further 1/8 loss in probability so that these two events do not occur either. Thus with probability at least 1/2, $p = p_+$, $q = p_-$ is a feasible solution to the regression problem and the value of the objective is

$$\begin{aligned}&\frac{1}{m}\sum_{i:\ y^i=1} \big|1 - p_+\big(x^i\big)\big| + \frac{1}{m}\sum_{i:\ y^i=0} \big|p_-\big(x^i\big)\big| \\ &\leqslant \frac{1}{m}\sum_{i:\ y^i=1} \big(\big|1 - c_+\big(x^i\big)\big| + \big|c_+\big(x^i\big) - p_+\big(x^i\big)\big|\big) + \frac{1}{m}\sum_{i:\ y^i=0} \big(\big|c_-\big(x^i\big)\big| + \big|c_-\big(x^i\big) - p_-\big(x^i\big)\big|\big) \\ &\leqslant \text{false}_-(c_+) + \text{false}_+(c_-) + \epsilon/2. \qquad \square\end{aligned}$$

Proof of Theorem 4. We use threshold functions θ_t in our proof, where for any $t \in [0,1]$, $\theta_t : \mathbb{R} \to \{0,1\}$ is defined as

$$\theta_t(x) = \begin{cases} 0 & \text{if } x < t, \\ 1 & \text{if } x \geqslant t. \end{cases}$$

Although the threshold functions θ_t do not occur in the algorithm, they significantly simplify the proof. To get a decision using the randomized hypothesis that our algorithm outputs, a simple technique is to choose a threshold uniformly in $[0,1]$ and use that to output a value in $\{0,1,?\}$. Thresholds over polynomials are halfspaces in a higher (n^d)-dimensional space, and we can use standard results from VC theory to bound the difference between the empirical error rates and true error rates. We will use frequently the following useful observation: For any $z \in \mathbb{R}$ we have

$$\text{crop}(z) = \int_{t=0}^{1} \theta_t(z)\, dt.$$

For a degree d polynomial $p : \mathbb{R}^n \to \mathbb{R}$, the function $\theta_t \circ p$ can be viewed as a halfspace in n^d dimensions, where we extend the terms of the polynomial to a linear function. Using the classical VC theory, for any distribution μ over $\mathcal{B}_n$, there exists a polynomial Q such that when $m = Q(n^d, \epsilon^{-1}, \delta^{-1})$, for a sample Z of m examples drawn from $D = (\mu, f)$, with probability at least $1 - \delta$, $|\widehat{\text{err}}(\theta_t \circ p, Z) - \text{err}(\theta_t \circ p)| \leqslant \epsilon$, where $\widehat{\text{err}}(g, Z) = \frac{1}{m}\sum_{(x,y)\in Z} I(g(x) \neq y)$. Similar bounds hold for false_+ and false_-, where $\widehat{\text{false}}_+(g, Z) = \frac{1}{m}\sum_{(x,y)\in Z} g(x)(1-y)$ and $\widehat{\text{false}}_-(g, Z) = \frac{1}{m}\sum_{(x,y)\in Z} (1 - g(x))y$.

Let $c = (c_+, c_-)$ be the best linear sandwich classifier with respect to tolerance rates τ_+, τ_-. Using results from Kalai, Klivans, Mansour and Servedio [12], for $d = O(1/\epsilon^4)$ there exist polynomials p_+, p_- such that $\mathbf{E}_{\mathcal{U}}[|p_+(x) - c_+(x)|] \leqslant \epsilon/128$ and $\mathbf{E}_{\mathcal{U}}[|p_-(x) - c_-(x)|] \leqslant \epsilon/128$. Thus when algorithm Sandwich-Learner is run with $T = \log\frac{2}{\delta}$ by Lemma 8, with probability at least $1 - \delta/2$, at least one of the iterations has value of the objective function smaller than $\text{false}_-(c_+) + \text{false}_+(c_-) + \epsilon/2$. It can be checked that $\text{false}_-(c_+) + \text{false}_+(c_-) = 1 - \text{acc}(c) = \text{err}(c) + ?(c)$. We assume that we are in this case where the least objective function is smaller than $1 - \text{acc}(c) + \epsilon/2$, allowing our algorithm to fail with probability $\delta/2$ so far. Let p, q be the polynomials which are solutions to the regression problem from the iteration with the least value of the objective function.

We now analyze the quantities false_+, false_-, acc of the randomized hypothesis h output by algorithm Sandwich-Learner. We present the analysis of false_+, false_- can be done similarly. We assume that the number of examples m is large enough, so that all the bounds due to VC theory that we require in the proof hold simultaneously with probability at least $1 - \frac{\delta}{2}$.[5] This can be done easily by taking union bound using only polynomial number of examples, say $P(n^d/(\epsilon\delta))$.

[5] All our requirements would be about comparing the error of a hyperplane to its observed error on the sample. We would need that the sample $Z = \{(x^1, y^1), \ldots, (x^m, y^m)\}$ is such that no hyperplane would have a difference larger than $\epsilon/8$.

For the first part, consider hyperplanes of the form $\theta_t \circ p$ and $\theta_t \circ (1-q)$, for any $t \in [0,1]$. By the VC bound we have that $|\widehat{\text{false}}_+(\theta_t \circ p, Z) - \text{false}_+(\theta_t \circ p, (\mathcal{U}, f))| \leqslant \epsilon/2$ and $|\widehat{\text{false}}_-(\theta_t \circ (1-q), Z) - \text{false}_-(\theta_t \circ (1-q), (\mathcal{U}, f))| \leqslant \epsilon/2$. The analysis then proceeds as follows:

$$
\begin{aligned}
\text{false}_+\big(h, (\mathcal{U}, f)\big) &= \mathop{\mathbf{E}}_{x\sim\mathcal{U}}\big[h(x,1)\big(1-f(x)\big)\big] \\
&= \mathop{\mathbf{E}}_{x\sim\mathcal{U}}\left[\int_{t=0}^{1} \theta_t\big(\min\big(p(x), q(x)\big)\big)\,dt\,\big(1-f(x)\big)\right] \\
&\leqslant \int_{t=0}^{1} \mathop{\mathbf{E}}_{x\sim\mathcal{U}}\big[\theta_t\big(p(x)\big)\big(1-f(x)\big)\big]\,dt \\
&= \int_{t=0}^{1} \text{false}_+\big(\theta_t \circ p, (\mathcal{U}, f)\big)\,dt \\
&\leqslant \int_{t=0}^{1} \widehat{\text{false}}_+(\theta_t \circ p, Z)\,dt + \frac{\epsilon}{2} \\
&= \int_{t=0}^{1} \frac{1}{m} \sum_{i:\ y^i=0} \theta_t\big(p\big(x^i\big)\big)\,dt + \frac{\epsilon}{2} \\
&\leqslant \frac{1}{m} \sum_{i:\ y^i=0} \big|p\big(x^i\big)\big| + \frac{\epsilon}{2} \\
&\leqslant \tau_+ + \epsilon \quad \big(\text{using constraint (9) from the regression problem}\big).
\end{aligned}
$$

Next we analyze the quantity $1 - \text{acc}(h) = \text{err}(h) + ?(h)$. Let $\bar{Z} = \{(x^1, 1-y^1), \ldots, (x^m, 1-y^m)\}$, namely we flip the labels in Z, and $Z^0 = \{(x^1, 0), \ldots, (x^m, 0)\}$, namely the sample Z with all labels 0. Here we assume the following bounds hold $|\widehat{\text{false}}_+(\theta_t \circ (1-p), \bar{Z}) - \text{false}_+(\theta_t \circ (1-p), (\mathcal{U}, 1-f))| \leqslant \epsilon/8$, $|\widehat{\text{false}}_+(\theta_t \circ q, Z) - \text{false}_+(\theta_t \circ q, (\mathcal{U}, f))| \leqslant \epsilon/8$ and $|\widehat{\text{err}}(\theta_t \circ (p-q), Z^0) - \text{err}(\theta_t \circ (p-q), (\mathcal{U}, 0))| \leqslant \epsilon/8$.

Step (16) below holds because $1 - \text{crop}(a) = \text{crop}(1-a)$. Step (17) uses the fact that $\min(a,b) = a - (a-b)\text{I}(a > b)$ and $\max(a,b) = b + (a-b)\text{I}(a > b)$. In step (18) we use $\text{crop}(a+b) \leqslant \text{crop}(a) + \text{crop}(b)$. All these facts can be checked easily.

$$
\begin{aligned}
1 - \text{acc}(h) &= 1 - \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[h(x,1)f(x) + h(x,0)\big(1-f(x)\big)\big] \\
&= \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\big(1-h(x,1)\big)f(x) + \big(1-h(x,0)\big)\big(1-f(x)\big)\big] \\
&= \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\big(1-\text{crop}\big(\min\big(p(x), q(x)\big)\big)\big)f(x) + \big(1-\text{crop}\big(\min\big(1-p(x), 1-q(x)\big)\big)\big)\big(1-f(x)\big)\big] \\
&= \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\text{crop}\big(1-\min\big(p(x), q(x)\big)\big)f(x) + \text{crop}\big(\max\big(p(x), q(x)\big)\big)\big(1-f(x)\big)\big] \qquad (16) \\
&= \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\text{crop}\big(1-p(x) + \big(p(x)-q(x)\big)\text{I}\big(p(x) > q(x)\big)\big)f(x) \\
&\quad + \text{crop}\big(q(x) + \big(p(x)-q(x)\big)\text{I}\big(p(x) > q(x)\big)\big)\big(1-f(x)\big)\big] \qquad (17) \\
&\leqslant \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\text{crop}\big(1-p(x)\big)f(x) + \text{crop}\big(q(x)\big)\big(1-f(x)\big) + \text{crop}\big(p(x)-q(x)\big)\big] \qquad (18) \\
&= \mathop{\mathbf{E}}_{x\in\mathcal{U}}\left[\int_{t=0}^{1} \theta_t\big(1-p(x)\big)f(x)\,dt + \int_{t=0}^{1} \theta_t\big(q(x)\big)\big(1-f(x)\big)\,dt + \int_{t=0}^{1} \theta_t\big(p(x)-q(x)\big)\cdot 1\,dt\right] \\
&= \int_{t=0}^{1} \Big(\mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\theta_t\big(1-p(x)\big)f(x)\big] + \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\theta_t\big(q(x)\big)\big(1-f(x)\big)\big] + \mathop{\mathbf{E}}_{x\in\mathcal{U}}\big[\theta_t\big(p(x)-q(x)\big)\big]\Big)\,dt \\
&= \int_{t=0}^{1} \big(\text{false}_+\big(\theta_t \circ (1-p), (\mathcal{U}, 1-f)\big) + \text{false}_+\big(\theta_t \circ q, (\mathcal{U}, f)\big) + \text{err}\big(\theta_t \circ (p-q), (\mathcal{U}, 0)\big)\big)\,dt
\end{aligned}
$$

$$
\begin{aligned}
&\leqslant \int_{t=0}^{1} \left(\widehat{\text{false}}_+\left(\theta_t \circ (1-p), \bar{Z}\right) + \widehat{\text{false}}_+(\theta_t \circ q, Z) + \widehat{\text{err}}\left(\theta_t \circ (p-q), Z^0\right)\right) dt + \frac{3\epsilon}{8} \\
&= \int_{t=0}^{1} \left(\frac{1}{m} \sum_{i:\ y^i=1} \theta_t\left(1-p\left(x^i\right)\right) + \frac{1}{m} \sum_{i:\ y^i=0} \theta_t\left(q\left(x^i\right)\right) + \frac{1}{m} \sum_{i=1}^{m} \theta_t\left(p\left(x^i\right) - q\left(x^i\right)\right)\right) dt + \frac{3\epsilon}{8} \\
&= \frac{1}{m} \sum_{i:\ y^i=1} \int_{t=0}^{1} \theta_t\left(1-p\left(x^i\right)\right) dt + \frac{1}{m} \sum_{i:\ y^i=0} \int_{t=0}^{1} \theta_t\left(q\left(x^i\right)\right) dt + \frac{1}{m} \sum_{i} \int_{t=0}^{1} \theta_t\left(p\left(x^i\right) - q\left(x^i\right)\right) dt + \frac{3\epsilon}{8} \\
&\leqslant \frac{1}{m} \sum_{i:\ y^i=1} \left|1-p\left(x^i\right)\right| + \frac{1}{m} \sum_{i:\ y^i=0} \left|q\left(x^i\right)\right| + \frac{1}{m} \sum_{i=1}^{m} \max\left(p\left(x^i\right) - q\left(x^i\right), 0\right) + \frac{3\epsilon}{8} \\
&\leqslant 1 - \text{acc}(c) + \epsilon
\end{aligned}
$$

where in the last inequality we used that fact that the first two terms are the objective function of the regression (and we assumed that they are at most $\text{false}_-(c_+) + \text{false}_+(c_-) + \epsilon/2$), and the third term is bounded in the regression by $\epsilon/8$. Hence $\text{acc}(h) \geqslant \text{acc}(c) - \epsilon$. □

References

[1] A. Cannon, J. Howse, D. Hush, C. Scovel, Learning with the Neyman–Pearson and min–max criteria, Technical Report LA-UR-02-2951, Los Alamos National Laboratory, 2002.

[2] P. Domingos, Metacost: A general method for making classifiers cost-sensitive, in: KDD'99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 1999, pp. 155–164.

[3] C. Elkan, The foundations of cost-sensitive learning, in: IJCAI'01: Proceedings of the 17th International Joint Conference on Artificial Intelligence, 2001, pp. 973–978.

[4] V. Feldman, Distribution-specific agnostic boosting, in: Proceedings of the First Symposium on Innovations in Computer Science'10, 2010.

[5] C. Ferri, P. Flach, J. Hernández-Orallo, Delegating classifier, in: ICML'04: Proceedings of the Twenty-First International Conference on Machine Learning, ACM, New York, NY, USA, 2004, pp. 37–44.

[6] C. Ferri, J. Hernández-Orallo, Cautious classifiers, in: Proceedings of First International Workshop on the ROC Analysis in Artificial Intelligence, 2004, pp. 27–36.

[7] O. Goldreich, L.A. Levin, A hard-core predicate for all one-way functions, 1989, pp. 25–32.

[8] P. Gopalan, A.T. Kalai, A.R. Klivans, Agnostically learning decision trees, in: STOC'08: Proceedings of the Fortieth Annual ACM Symposium on the Theory of Computation, ACM, New York, NY, USA, 2008.

[9] D. Haussler, Decision-theoretic generalizations of the PAC model for neural networks and other applications, Inform. and Comput. 100 (1992) 78–150.

[10] J.C. Jackson, An efficient membership-query algorithm for learning DNF with respect to the uniform distribution, J. Comput. System Sci. 55 (3) (1997) 414–440.

[11] A. Kalai, V. Kanade, Potential-based agnostic boosting, Adv. Neural Inf. Process. Syst. 22 (2009) 880–888.

[12] A.T. Kalai, A.R. Klivans, Y. Mansour, R.A. Servedio, Agnostically learning halfspaces, in: FOCS'05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, USA, 2005, pp. 11–20.

[13] M.J. Kearns, R.E. Schapire, L.M. Sellie, Toward efficient agnostic learning, Mach. Learn. 17 (2) (1994) 115–141.

[14] E. Kushilevitz, Y. Mansour, Learning decision trees using the Fourier spectrum, SIAM J. Comput. 22 (6) (1993) 1331–1348.

[15] J. Neyman, E.S. Pearson, On the problem of the most efficient tests for statistical hypotheses, Philos. Trans. R. Soc. Lond. Ser. A Contain. Pap. Math. Phys. Character 231 (1933) 289–337.

[16] Tadeusz Pietraszek, On the use of ROC analysis for the optimization of abstaining classifiers, Mach. Learn. 68 (2) (2007) 137–169.

[17] C. Scott, R. Nowak, A Neyman–Pearson approach to statistical learning, IEEE Trans. Inform. Theory 51 (11) (2005) 3806–3819.

[18] L.G. Valiant, A theory of the learnable, Commun. ACM 27 (11) (1984) 1134–1142.

[19] B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in: ICDM'03: Proceeding of the Third IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2003, p. 435.