Learning Reliable Rules under Class Imbalance*

Dimitrios I. Diochnos† Theodore B. Trafalis†

Abstract

We are interested in learning rules under class imbalance. In this direction we extend the traditional model of probably approximately correct (PAC) learning to also include explicitly among its goals high recall and high precision at the end of the learning process. We establish relationships for the recall and the precision of a learned hypothesis as a function of its risk and the rate of the minority class. We then show that we can PAC learn a concept class with high recall and high precision by allowing a polynomial increase in the time and space complexity of traditional PAC learning algorithms that generate hypotheses with low risk. In sequence, by introducing a pre-processing phase on such algorithms, with a constant-size overhead on the overall sample complexity, we are able, with high probability, to compute a lower bound of the true unknown rate p of the minority class, in the interval $[p/8, p)$. Thus, we extend our positive results on PAC learning with high recall and high precision by also waiving the requirement that such a lower bound on the rate of the minority class is given to the learners by some oracle ahead of time. We conclude our work by exploring two popular PAC learning algorithms for monotone conjunctions.

1 Introduction

In the last 40 years data mining and machine learning have grown to be very rich research fields with many important theoretical results and a tremendous amount of successful applications that have changed our daily lives. During this time of progress, theory and practice oftentimes have complemented each other and have caused further inspiration and motivation for additional results in both directions.

From a theoretical point of view, perhaps the most popular learning model is that of the *probably approximately correct (PAC)* model of learning [30]. The goal in PAC learning is for the learner to come up with a rule, that, with high probability, has low risk (error rate) on unseen data of some underlying problem of interest. In this model several concept classes have been studied and while the original purpose of the model was the production of distribution-independent results, there is also a plethora of results relevant to distribution-specific learning where one studies the guarantees of certain learning algorithms under specific distributions, as is, for example, a multivariate Gaussian distribution in $\mathbb{R}^n$, a product distribution on the Boolean hypercube, or other distributions that have properties that are appealing to human (mathematical) intuition.

On the other hand, in several practical situations we have the issue of *class imbalance* [6,11,19,27], where the class that we want to learn is under-represented in the data that we have access to – this poses additional difficulties on learning algorithms. For example, some algorithms expect more samples (higher ratio) from the class of interest so that they can learn a good predictive model. Moreover, we might be interested in learning *rare events* that occur with probability between 0.02% to 25%. There are numerous disparate situations along these lines; e.g., anomaly detection [15], e-commerce [3], weather phenomena [20,28], and many others. However, especially in the cases of *extreme class imbalance* [16] where the minority class may occur with probability 0.02%-0.1%, generating a model that achieves low risk is not difficult at all, since, one can always predict that the rare event will not happen, and such predictions will be highly accurate. In these situations we are interested in more detailed information on the performance of the models that we learn. The relevant metrics are *recall* (measuring the rate of correct predictions among rare events alone) and *precision* (measuring how accurate our model is when it predicts a rare event), as well as combinations of these two, such as the F_1 score (harmonic mean), or the G-measure (geometric mean).

Our work in this paper is motivated by situations where we would like the learnt hypothesis to have not only low risk (error rate) but also to be *reliable* and satisfy such more delicate metrics as the ones that are relevant for rare events; i.e., high recall and high precision. We do so by extending the traditional framework for PAC learning.

1.1 Related Work

Many different approaches have been designed and are applied in practice in situations where we have class imbalance. Such approaches in-

*This material is based upon work supported by the National Science Foundation under Grant No. ICER-2019758. This work is part of the NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography (AI2ES).

†University of Oklahoma

clude, random under-sampling the majority class or random over-sampling the minority class, informed under-sampling the majority class [18], the creation of synthetic data for helping over-sampling [8], sampling based on clusters [12], re-weighting [33], as well as methods that rely on well-established notions and techniques such as dealing with the margin of the separation boundary [7], modifying the solution obtained through support vector machines [34], using boosting [26], and a lot of other approaches that inevitably we cannot list due to space limitations. These methods are primarily attempting to address the issue of *absolute class imbalance* that exists on specific datasets.

However, *relative class imbalance* is still an important issue in datasets where we have an abundance of training examples, but in which the distribution of the different classes might be severely skewed. It is this latter situation that we are interested in this paper, where one can have access to enough examples from the minority class, even if the frequency of the minority class is very small, as long as the total number of examples is sufficiently big. For this reason we will embed these requirements into the definition of PAC learning explicitly. During the submission the only work that we were aware of that provided generalization guarantees on the minority class, in a probabilistic sense, was the work of [7]. Another direction that was pointed out to us by one of the reviewers was that of learning algorithms that satisfy *complex performance measures*; i.e., algorithms whose goal is to optimize arbitrary functions on the entries of the confusion matrix, e.g., [13]; where we may also find probabilistic results in the spirit of our proposed framework, e.g., [23].

1.2 Contributions and Organization of the Paper In Section 2 we present basic definitions, the traditional model of PAC learning and some known complexity results (referring to realizable learning problems).

In Section 3 we extend PAC learning to situations where we care about learning rare events. This formalization is our first contribution as such an extension, to the best of our knowledge, is missing from the literature and can motivate further work on algorithms that not only generate hypotheses (models/rules) that have low risk but also provide guarantees on their recall and precision.

In Section 3.1, our second contribution is to provide lower bounds on the recall and precision of a learned hypothesis by bringing together the risk of the hypothesis as well as the rate of the minority class. While the lower bound on the recall depends on only these two parameters, the lower bound for the precision also depends on (a lower bound of) the recall. Using these lower bounds on recall and precision, we then show in Section 3.2 that in order to achieve PAC learning with recall at least $1-\gamma$ and precision at least $1-\xi$, it is enough to learn a hypothesis to risk at most $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$, where ε is the usual risk bound that we want to achieve and p_b is a lower bound on the rate of the minority class.

The above result assumes that a lower bound p_b on the rate of the minority class is given to the learner. In Section 3.3 our next contribution is to require a 'preprocessing' step, that allows us to waive this requirement of a priori knowledge, by finding, with high probability, an appropriate lower bound p_b on the rate of the minority class through a constant-size overhead of training examples.

In Section 4 we explore experimentally two algorithms for monotone conjunctions and study the quality of the generated solutions in the traditional, as well as in the proposed, PAC framework.

We conclude our work in Section 5 with a summary, as well as ideas for future work.

Omitted discussion and proofs are available in the supplementary material.

2 Preliminaries

We study binary classification problems. We denote with 1 the label of positive examples and with 0 the label of negative examples. With $\mathbf{1}_{\mathcal{A}}$ we denote the indicator function of a probability event $\mathcal{A}$. That is, $\mathbf{1}_{\mathcal{A}}$ is 1 when the event $\mathcal{A}$ holds, whereas $\mathbf{1}_{\mathcal{A}}$ is 0 when the event $\mathcal{A}$ does not hold. We will use the definition for the conditional probability of two events; i.e., if $\mathcal{A}$ and $\mathcal{B}$ are two probability events such that $\mathbf{Pr}(\mathcal{B}) > 0$, we have $\mathbf{Pr}(\mathcal{A} \mid \mathcal{B}) = \mathbf{Pr}(\mathcal{A} \cap \mathcal{B}) / \mathbf{Pr}(\mathcal{B})$. With $\lg(a)$ we denote the logarithm of a in base 2, whereas with $\ln(a)$ we denote the natural logarithm of a. With $\log_b(c)$ we denote the logarithm of c in base b. We use the $\wedge$ connective to denote conjunctions (logical AND functions).

2.1 Probably Approximately Correct Learning The probably approximately correct (PAC) model of learning that was introduced by Valiant in [30] is perhaps the most well-studied and most extensively used learning framework. Learning problems in this framework can be specified as tuples of the form $(\mathcal{X}, \mathcal{C}, \mathcal{H}, \mathcal{D})$, where $\mathcal{X}$ is the set of *instances*, $\mathcal{C} \subseteq 2^{\mathcal{X}}$ is the *concept class*, $\mathcal{H} \subseteq 2^{\mathcal{X}}$ is the *hypothesis space*, and $\mathcal{D}$ is a set of distributions over $\mathcal{X}$. The goal of the learner is to be able to learn *approximately* an unknown function c that is drawn from the known concept class $\mathcal{C}$, using the functions that are available in the hypothesis space $\mathcal{H}$, and such a guarantee can be provided with *high probability*, regardless of the underlying distribu-

tion $D \in \mathcal{D}$ that is induced over $\mathcal{X}$. This is why we have the name of *probably approximately correct* learning and we will start formalizing these notions in the following paragraph. When the set of distributions $\mathcal{D}$ can be any distribution over $\mathcal{X}$, we talk about *distribution-free* (or *distribution-independent*) learning and in fact in this case learning problems are usually specified using the triple $(\mathcal{X}, \mathcal{C}, \mathcal{H})$, where now $\mathcal{D}$ is omitted from the original tuple. In the other extreme, we can have the case that $\mathcal{D}$ is a singleton, e.g., the only distribution found in $\mathcal{D}$ is the uniform distribution over $\{0,1\}^n$, where we now discuss about *distribution-specific* learning. In between these two extremes one can still allow lots of diversity and study learning problems; e.g., $\mathcal{D}$ is the set of all product distributions over $\{0,1\}^n$.

Learning occurs by drawing a sample S of m instances $(x_1, \ldots, x_m)$ i.i.d. according to $D \in \mathcal{D}$ and then each one of those instances is labeled by the unknown concept $c \in \mathcal{C}$ giving $(c(x_1), \ldots, c(x_m))$. These two sequences are then put together in terms of instance-label pairs, and provide the tuple $T = ((x_1, y_1), \ldots, (x_m, y_m))$ which contains m *training examples*, where for each label y_i with $i \in \{1, \ldots, m\}$, it holds $y_i = c(x_i)$. These training examples T are presented to the learner and the learner now has to identify a *hypothesis* $h \in \mathcal{H}$ that has *low risk* against c when evaluated under the distribution D; i.e., learning an h and evaluating h occurs under the same distribution. This evaluation is measured by the *risk* of the learnt hypothesis h and is defined as follows.

Definition 2.1. (Risk) *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution D, the* risk *of h is defined by*

$$R_D(h, c) = \mathbf{Pr}_{x \sim D}(h(x) \neq c(x)) = \mathbf{E}_{x \sim D}\left[\mathbf{1}_{h(x) \neq c(x)}\right].$$

The underlying distribution $D \in \mathcal{D}$ may, or may not, be known to the learner - for example, it is always the case that the learner does not know the underlying distribution when we discuss PAC learning results in a distribution-free sense. In any case however, the labeling function c is always unknown to the learner as this is the function that one wants to approximate well. As a consequence, the risk of any hypothesis $h \in \mathcal{H}$ is not directly known to the learner since it involves at least one unknown quantity. Thus, the learner is usually using as a proxy the *empirical risk* of a hypothesis.

Definition 2.2. (Empirical Risk) *Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and a sample $S = (x_1, \ldots, x_m)$, the* empirical risk *of h is defined by*

$$\widehat{R}_S(h, c) = \frac{1}{m} \cdot \sum_{i=1}^{m} \mathbf{1}_{h(x_i) \neq c(x_i)}.$$

As learning ultimately has to deal with representations, we typically use n for the dimension of the instances; e.g., each instance $x \in \mathcal{X}$ is a truth assignment in $\{0,1\}^n$, or a vector in $\mathbb{R}^n$. Furthermore, with $size(c)$ we denote the computational cost for representing the true target c that we want to learn. We are now ready to define PAC learning.

Definition 2.3. (PAC Learning) *A concept class $\mathcal{C}$ is said to be PAC-learnable by a hypothesis class $\mathcal{H}$, if there exists a learning algorithm A and a polynomial function $poly(\cdot, \cdot, \cdot, \cdot)$, such that for any $\varepsilon > 0$ and $\delta > 0$, for all distributions $D \in \mathcal{D}$ over $\mathcal{X}$, for any target concept $c \in \mathcal{C}$, for any sample S of size $m \geq poly(1/\varepsilon, 1/\delta, n, size(c))$, algorithm A outputs a hypothesis $h \in \mathcal{H}$, such that:*

$$\mathbf{Pr}_{S \sim D^m}(R_D(h, c) \leq \varepsilon) \geq 1 - \delta.$$

Furthermore, if the algorithm A runs in time $poly(1/\varepsilon, 1/\delta, n, size(c))$, then $\mathcal{C}$ is said to be efficiently *PAC-learnable by the hypothesis class $\mathcal{H}$.*

For a broader, but still brief, exposition of PAC learning and related models and results, a good source is [29]. However, there is a plethora of recent and older books around PAC learning and the interested reader can find lots of useful information there; e.g., [14,22,25].

2.2 Known Results Several results in the framework of PAC learning are obtained using the realizability assumption.

Definition 2.4. (Realizable Learning Problem) *A learning problem $(\mathcal{X}, \mathcal{C}, \mathcal{H}, \mathcal{D})$ is said to be realizable, if for any $D \in \mathcal{D}$ and any $c \in \mathcal{C}$, there exists at least one $h \in \mathcal{H}$ such that $R_D(h, c) = 0$.*

Theorem 2.1. ([4]) *Let $\mathcal{H}$ be a finite hypothesis class. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity $m \leq \left\lceil \frac{1}{\varepsilon} \cdot \ln\left(\frac{|\mathcal{H}|}{\delta}\right) \right\rceil$.*

The Vapnik-Chervonenkis dimension VC-dim$(\mathcal{H})$, is a combinatorial parameter that characterizes the richness of a class of functions $\mathcal{H}$ [32]. In particular, VC-dim$(\mathcal{H})$ is the size of the largest set of instances such that for any possible labeling on these instances, there is at least one function in $\mathcal{H}$ that admits the particular labeling. This parameter allows us to prove an analog of Theorem 2.1 when $|\mathcal{H}| = \infty$ but VC-dim$(\mathcal{H}) = d < \infty$.

Theorem 2.2. ([5]) *Let $\mathcal{H}$ be a hypothesis class with VC-dim$(\mathcal{H}) = d < \infty$. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity $m \leq \left\lceil \frac{4}{\varepsilon} \cdot (d \lg(12/\varepsilon) + \lg(2/\delta)) \right\rceil$.*

The constants in Theorem 2.2 are obtained from [2].

3 PAC Learning with High Recall and High Precision

Throughout the rest of the paper we use the label 1 for the label of the minority class. In order to address the issues of class imbalance we want to learn a hypothesis that not only has low risk, but also has high recall and/or high precision. However, dealing with imbalanced data implies that we are working under the assumption of $\mathbf{Pr}_{x\sim D}(c(x)=1)>0$ and hence we can have the following definition.

DEFINITION 3.1. (RECALL) *Given a hypothesis $h\in\mathcal{H}$, a target concept $c\in\mathcal{C}$, and an underlying distribution D, the* recall *of h is defined by*

$$\text{REC}_D(h,c)=\mathbf{Pr}_{x\sim D}(h(x)=1\mid c(x)=1)\ .$$

Similarly, for a hypothesis h where we have the guarantee that $\mathbf{Pr}_{x\sim D}(h(x)=1)>0$, we can define the following.

DEFINITION 3.2. (PRECISION) *Given a hypothesis $h\in\mathcal{H}$, a target concept $c\in\mathcal{C}$, and an underlying distribution D, the* precision *of h is defined by*

$$\text{PREC}_D(h,c)=\mathbf{Pr}_{x\sim D}(c(x)=1\mid h(x)=1)\ .$$

We can now introduce the constraints of high recall and high precision into Definition 2.3.

DEFINITION 3.3. (PAC LEARNING EXTENSION) *A concept class $\mathcal{C}$ is said to be PAC-learnable with high recall and high precision by a hypothesis class $\mathcal{H}$, if there exists a learning algorithm A and a polynomial function $poly(\cdot,\cdot,\cdot,\cdot,\cdot,\cdot)$, such that for any $\varepsilon>0$, $\delta>0$, $\gamma>0$, and $\xi>0$, for all distributions $D\in\mathcal{D}$ over $\mathcal{X}$, for any target concept $c\in\mathcal{C}$, for any sample $\mathcal{S}$ of size $m\geq poly(1/\epsilon,1/\delta,1/\gamma,1/\xi,n,size(c))$, algorithm A outputs a hypothesis $h\in\mathcal{H}$, such that:*

$$\mathbf{Pr}_{S\sim D^m}\left(\begin{array}{rcl}(R_D(h,c) & \leq & \varepsilon)\\ \wedge\ (\text{REC}_D(h,c) & \geq & 1-\gamma)\\ \wedge\ (\text{PREC}_D(h,c) & \geq & 1-\xi)\end{array}\right)\geq 1-\delta\,.$$

Furthermore, if the algorithm A runs in time $poly(1/\epsilon,1/\delta,1/\gamma,1/\xi,n,size(c))$, then $\mathcal{C}$ is said to be efficiently *PAC-learnable with high recall and high precision by the hypothesis class $\mathcal{H}$.*

In Definition 3.3 one can drop the reference to the notion of recall, or to the notion of precision, and obtain a version of PAC learning where one focuses only on precision, or only on recall, respectively. Of course, by dropping the dependence on both the recall and the precision we obtain Definition 2.3, which is about the traditional variant of PAC learning.

3.1 Lower Bounds on Recall and Precision

The following propositions show that if we have a lower bound on the rate of the minority class, then we can obtain a lower bound on the recall and the precision of a hypothesis h by incorporating the risk $R_D(h,c)$ as well.

PROPOSITION 3.1. (LOWER BOUND FOR RECALL) *Let p_b be given such that $\mathbf{Pr}_{x\sim D}(c(x)=1)\geq p_b>0$. Let $h\in\mathcal{H}$ be a hypothesis with risk $R_D(h,c)$. Then, for this hypothesis h it holds*

$$\text{REC}_D(h,c)\geq 1-\frac{R_D(h,c)}{p_b}\,.$$

PROPOSITION 3.2. (LOWER BOUND FOR PRECISION) *Let p_b be given such that $\mathbf{Pr}_{x\sim D}(c(x)=1)\geq p_b>0$. Let $h\in\mathcal{H}$ be a hypothesis with risk $R_D(h,c)$ and for which it holds $\text{REC}_D(h,c)\geq 1-\gamma$ for some $0\leq\gamma<1$. Then, for this hypothesis h it holds*

$$\text{PREC}_D(h,c)\geq 1-\frac{R_D(h,c)}{(1-\gamma)p_b}\,.$$

3.2 Implications

Having the lower bounds of Section 3.1 we can now prove a positive result for PAC learning with high recall and high precision.

THEOREM 3.1. *Let L be a learner such that, for every $0<\varepsilon,\delta<1$, L can produce an $h\in\mathcal{H}$ that achieves the PAC criterion (Definition 2.3) when learning $c\in\mathcal{C}$ using hypotheses from $\mathcal{H}$ under a set of distributions $\mathcal{D}$ over $\mathcal{X}$. Let p_b be an input parameter that is known to the learner such that $\mathbf{Pr}_{x\sim D}(c(x)=1)\geq p_b>0$. Then, for any $0<\xi<1$ and any $0<\gamma\leq 1/2$, using L to generate an $h\in\mathcal{H}$ for which it holds $R_D(h,c)\leq\min\{\varepsilon,\gamma p_b,\xi p_b/2\}$ implies for the same h that $\text{REC}_D(h,c)\geq 1-\gamma$ as well as $\text{PREC}_D(h,c)\geq 1-\xi$. That is, L PAC learns $\mathcal{C}$ with high recall and high precision using $\mathcal{H}$.*

Theorem 3.1 refers to situations where we may or may not be dealing with realizable learning problems. For realizable learning problems, by combining Theorem 3.1 and Theorem 2.2, we can obtain the following.

COROLLARY 3.1. *Let $\mathcal{H}$ be a hypothesis class with VC-dim$(\mathcal{H})=d<\infty$. Let p_b be an input parameter that is known to the learner such that $\mathbf{Pr}_{x\sim D}(c(x)=1)\geq p_b>0$. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable with high recall and high precision by $\mathcal{H}$ with sample complexity $m\leq\left\lceil\frac{4}{r}\left(d\lg\left(\frac{12}{r}\right)+\lg\left(\frac{2}{\delta}\right)\right)\right\rceil$. where, this time, $r=\min\{\varepsilon,\gamma p_b,\xi p_b/2\}$.*

Corollary 3.1 above is presented using the VC-dimension so that we can provide statements and bounds for the more general case, as for every finite hypothesis space $\mathcal{H}$, VC-dim $(\mathcal{H}) \leq \lg(\mathcal{H})$. Of course one can prove an analog of Corollary 3.1 for finite concept classes, by invoking Theorem 2.1 directly.

3.3 Lower Bound on the Minority Class Rate

Here we discuss how to compute a lower bound p_b for the quantity $\mathbf{Pr}_{x\sim D}(c(x)=1)$ so that we can waive the requirement that such a lower bound is provided to Theorem 3.1. Our idea for obtaining such a lower bound p_b is along the lines of bisecting an initial estimate, until we are confident, with high probability, that we have indeed computed the desired lower bound. Our approach resembles how [1, 17] identify an *upper bound* of the true, unknown, noise rate in the random misclassification noise model of PAC learning.

Briefly, in round 1 we guess that the minority class occurs with probability larger than 1/8. If our guess appears to be correct, we stop and use $p_b = 1/8$. Otherwise, in each round, we are halving our guess for the lower bound that corresponds to the minority class rate, until we are correct. Algorithm 1 has the details.

Algorithm 1 Computing a lower bound on the minority class rate.

```
i ← 1 and p_b ← 2^{-3}
while TRUE do
  Draw m_i ≥ ⌈2^{3+2i} ln(2^{1+i}/δ)⌉ examples and let k_i
  of those belong to the minority class
  p̂_i ← k_i/m_i
  if p̂_i ≤ 2^{-(1+i)} then
    i ← i + 1 and p_b ← p_b/2
  else
    break
  end if
end while
return p_b
```

Lemma 3.1. *Let $\mathbf{Pr}_{x\sim D}(c(x)=1) = p > 0$. Let $m_i \geq \lceil 2^{3+2i}\ln(2^{1+i}/\delta)\rceil$ for $i \in \{1,2,\ldots\}$. Then, with probability more than $1-\delta$, Algorithm 1 halts within $\lceil \lg(3/2p)\rceil$ iterations and provides a lower bound p_b such that $0 < p/8 \leq p_b < p$.*

Corollary 3.2. *Lemma 3.1 requires total sample size $\mathcal{O}\left(\frac{1}{p^2}\cdot\ln\left(\frac{1}{p\delta}\right)\right)$.*

Table 1 shows an upper bound on the sample size provided to Algorithm 1 that would be enough for successful termination, in various cases of the minority rate as well as on the probability of successful termination.

Table 1: Upper bound on the number of examples requested by Algorithm 1 (Lemma 3.1) in order to compute a lower bound, with high confidence, on the rate of the minority class.

Minority Rate (p)	**Confidence**		
	0.9	***0.95***	***0.99***
20%	13,693	15,356	19,219
10%	61,415	68,069	83,520
5%	272,264	298,881	360,684
1%	8,351,543	9,016,964	10,562,024
0.5%	36,067,831	38,729,516	44,909,758
0.1%	1,056,201,596	1,122,743,726	1,277,249,765
0.05%	4,490,974,869	4,757,143,386	5,375,167,545

We can now waive the requirement that a learner is fed with a lower bound p_b for the minority class rate p, as this will be computed using Algorithm 1 with associated failure probability $\delta/2$. Then we can use p_b and proceed as before but now requiring failure probability at most $\delta/2$ for the second part of the learning process. By the union bound, both phases terminate successfully except with probability at most δ. Also, since p_b satisfies $p_b \geq p/8$, we can argue about the statistical complexity of various learning methods by using directly the unknown probability $p = \mathbf{Pr}_{x\sim D}(c(x)=1)$. Below we extend Theorem 3.1.

Theorem 3.2. *Let L be a learner such that, for every $0 < \varepsilon, \delta < 1$, L can produce an $h \in \mathcal{H}$ that achieves the PAC criterion (Definition 2.3) when learning $c \in \mathcal{C}$ using hypotheses from $\mathcal{H}$ under a set of distributions $\mathcal{D}$ over $\mathcal{X}$. Using Algorithm 1, with $\mathcal{O}\left(\frac{1}{p^2}\ln\left(\frac{1}{p\delta}\right)\right)$ examples we can compute, except with probability at most $\delta/2$, a lower bound p_b of the true unknown rate p of the minority class, such that $p/8 \leq p_b < p$. Then, for any $0 < \xi < 1$ and any $0 < \gamma \leq 1/2$, we use L to generate, except with probability at most $\delta/2$, an $h \in \mathcal{H}$ for which it holds $R_D(h,c) \leq \min\{\varepsilon, \gamma p_b, \xi p_b/2\}$. For the generated hypothesis h it also holds that $\text{REC}_D(h,c) \geq 1-\gamma$ as well as $\text{PREC}_D(h,c) \geq 1-\xi$. That is, L PAC learns $\mathcal{C}$ with high recall and high precision using $\mathcal{H}$, even when a lower bound on the true rate of the minority class is not known in advance.*

4 Case Study: Monotone Conjunctions

One of the most fundamental classes of rules that have been explored in the theory and practice of data mining, is the class of rules that can be formed by taking a conjunction, without allowing negations, of some of the available (Boolean) variables. In terms of Boolean

functions, this class of rules is known as the class of *monotone conjunctions* and this is the concept class $\mathcal{C}$ that we want to study in this section. Below we may omit repeating the word *monotone*, but it is only monotone conjunctions that we study.

A function in $\mathcal{C}$ can be, for example in a space where we have $n \geq 4$ Boolean variables, $c = x_1 \wedge x_2 \wedge x_4$, indicating that $c(x) = 1$ if the first, second and fourth attribute of x are simultaneously all 1's; otherwise $c(x) = 0$. This particular c depends on 3 Boolean variables and for this reason we say that the *size* (or, the *length*) of c is 3 and we denote this by writing $|c| = 3$. Therefore, the concept class $\mathcal{C}$ contains $|\mathcal{C}| = 2^n$ monotone conjunctions since every subset of $\{x_1, \ldots, x_n\}$ corresponds to a different function in $\mathcal{C}$ and in fact this mapping is a bijection. Now consider the case where our hypothesis space is $\mathcal{H} = \mathcal{C}$ and let us compare an arbitrary $h \in \mathcal{H}$ against an arbitrary $c \in \mathcal{C}$ as shown below:

$$c = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{k=1}^{u} y_k \quad \text{and} \quad h = \bigwedge_{i=1}^{m} x_i \wedge \bigwedge_{\ell=1}^{w} z_\ell \,. \tag{4.1}$$

That is, there is a *mutual* part between h and c and beyond that, the two functions may include different variables. For this reason we will call the variables that appear in c but not in our hypothesis h as *undiscovered*, while we will call the variables that appear in h but not in c as *wrong*. Hence, in (4.1) there are m mutual variables, u undiscovered and w wrong. Also, we will call the variables that appear in c as *good*, otherwise *bad*. Based on the decomposition that is shown in (4.1) we can observe the following.

PROPOSITION 4.1. *Let D be a product distribution over $\{0,1\}^n$ where each variable is satisfied with the same probability λ. Consider a target c and a hypothesis h as in (4.1). Then,*

$$\begin{cases} R_D(h,c) &= \lambda^m \left(\lambda^u + \lambda^w - 2\lambda^{u+w}\right) \\ \textsc{Rec}_D(h,c) &= \lambda^w \\ \textsc{Prec}_D(h,c) &= \lambda^u \end{cases}$$

4.1 Two Algorithms We study the learnability of this class of functions $\mathcal{C}$ using two different learning methods: the algorithm that is known as FIND-S in [21], as well as the SWAPPING ALGORITHM from the framework of evolvability [31].

4.1.1 The Algorithm FIND-S FIND-S starts from the full monotone conjunction and as positive examples are received during training, the learner drops from the hypothesis those variables whose entries are zeros in the positive examples. Our hypothesis class is $\mathcal{H} = \mathcal{C}$ and therefore we are dealing with a realizable learning problem. During the learning process FIND-S maintains a hypothesis that is consistent with all the training examples seen thus far. Hence, in order to satisfy the PAC criterion (Definition 2.3), it is enough if the learner obtains m examples as m is dictated by either Theorem 2.1 or Theorem 2.2. In the first case (Theorem 2.1) we can use the fact that $\mathcal{H}$ is finite and in particular $|\mathcal{H}| = |\mathcal{C}| = 2^n$, while in the second case (Theorem 2.2) we can use the fact that VC-dim $(\mathcal{H}) = n$. Either way, it follows that a polynomial sample m is enough for PAC learning $\mathcal{C}$. Furthermore the algorithm is efficient as it processes each training example only once, potentially removing some variables that appear in the hypothesis as each example is being processed.

4.1.2 The SWAPPING ALGORITHM This algorithm performs a local-search approach on forming a hypothesis, by swapping in and out of the hypothesis some variable(s) as the correlation of the hypothesis with the target function is approximated based on training examples that become available to the learning process. See [10] for the uniform distribution over $\{0,1\}^n$ and [9] for product distributions where each variable is satisfied with the same probability $\lambda \in (0,1)$. As the results where largely similar for various values of the parameter λ, below we report only on the results obtained for the uniform distribution.

The hypothesis space of this algorithm is $\mathcal{H}_s = \left\{ f \mid f \in \mathcal{C} \text{ and } |f| \leq \left\lceil \log_{1/\lambda}(3/(2\varepsilon)) \right\rceil \right\}$, where λ characterizes the underlying probability distribution. Therefore, in particular, under the uniform distribution the generated hypotheses have at most $\lceil \lg(3/(2\varepsilon)) \rceil$ variables. The algorithm proceeds in iterations and as new batches of training examples are received, the hypothesis is updated by either adding a variable, or removing a variable, or swapping a variable for another variable, or finally by leaving the hypothesis unchanged. Briefly, the idea is that when $|c| \leq \left\lceil \log_{1/\lambda}(3/(2\varepsilon)) \right\rceil$, then c is identified precisely (so, $R_D(h,c) = 0$, and consequently $\textsc{Rec}_D(h,c) = \textsc{Prec}_D(h,c) = 1$), while if $|c| > \left\lceil \log_{1/\lambda}(3/(2\varepsilon)) \right\rceil$, then there is at least one $h \in \mathcal{H}_s$ for which it holds $R_D(h,c) \leq \varepsilon$ and the algorithm generates such a hypothesis at the end of the training process. Note that by the nature of the hypothesis space, this is not a realizable learning problem in the general case (and in particular in our setting). As such, requiring high recall and high precision on the solution, one now needs to invoke Theorem 3.1 or Theorem 3.2 directly, and not corollaries of the well-established Theorems 2.1 and 2.2. This contrasts FIND-S where one, for example, can invoke Corollary 3.1 directly.

4.2 Experiments For the experiments we skip the process of identifying a lower bound on the rate p of the minority class and we use the fact that at the end of such a process we will identify a lower bound p_b satisfying $p/8 \leq p_b < p$. Due to Theorem 3.1, in the extended version of PAC learning (Definition 3.3), we want to bound the risk by a quantity that depends linearly on p_b (and hence the sample size inversely depends on p_b), it follows that the closer p_b is to the true rate p, then the fewer examples are needed for achieving the same guarantee. Hence, even if Algorithm 1 technically computes a p_b strictly less than p, we will use $p_b = p$ so that the augmented sample size that is fed to the learning process, is as small as possible based on the results that we discussed earlier (e.g., Corollary 3.1).

Getting along, we fix the bound on the risk of the learned hypothesis to be at most 5%; that is, we set $\varepsilon = 0.05$. Also, we fix the confidence level to be $\delta = 0.1$. As imbalanced datasets typically range anywhere between 0.02% and 25%, we explore target functions that induce rates for the minority class in this range. Therefore, under the uniform distribution ($\lambda = 1/2$), we explore the cases where the target monotone conjunction has size $|c|$ in the set $\{2, 3, \ldots, 12\}$, which in turn implies rates p for the minority class in the set $\left\{2^{-2}, 2^{-3} \ldots, 2^{-12}\right\}$; that is, rates as large as 25% and as small as about 0.024%. In addition, we fix the dimension of the instances to be $n = 100$; that is, we work in a space where there are 100 Boolean variables. Moreover, we perform 1,000 experiments for each different minority class rate.

The source code is available at the following url:

`https://github.com/diochnos/pac-imbalanced`

4.2.1 Experiments with FIND-S It holds that $|\mathcal{H}| = 2^n = 2^{100}$. Hence, in order to PAC learn a hypothesis that has risk at most 0.05, due to Theorem 2.1, $m = \left\lceil \frac{1}{0.05} \left(100 \ln(2) + \ln(10)\right)\right\rceil = 1,433$ training examples are enough and this is the sample size that we use for FIND-S in the vanilla version of PAC learning (Definition 2.3). Table 2 has details on the solutions that we obtained. We can see that for targets with at most 6 variables ($p = 2^{-6} \approx 1.563\%$), the generated hypothesis identified the target precisely in each of the 100 runs. Beyond that point, we can see that the *worst case risk* of the generated hypothesis can be non-zero and in particular almost identical to the rate of the minority class. Furthermore, as FIND-S never deletes good variables from the hypothesis, the solutions obtained always have precision equal to 1 and hence we do not comment on the precision in Table 2.

We then performed experiments in the proposed framework of PAC learning with high recall and high precision. In this direction we set $\gamma = 0.4$ and $\xi = 0.9$. As we discussed earlier FIND-S generates solutions that have precision equal to 1, and for this reason we selected a high value for ξ, so that when we look at the quantity $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$, the term $\xi p_b/2$ is always larger than the term γp_b and as a result make the sample size depend only on ε or on the term γp_b. In all of our experiments, the solutions that we obtained identified the target precisely and as a result all the solutions had risk 0, recall 1 and precision 1. This happened even though γ was set in such a way so that we did not require a very large value on the recall. Therefore, the increase in the sample size due to the knowledge of a lower bound on the rate of the minority class this time resulted in the creation of perfect hypotheses in every run. Regarding the sample size, for the cases where $p = 25\%$ or $p = 12.5\%$ it holds that $\varepsilon \leq \gamma p_b$ and therefore the sample size there was $1,433$ as in the previous case. Then, as the minority rate gets halved, the number of examples double in each case, so we need $2,865$ examples when $p = 6.25\%$, $5,730$ examples when $p = 3.125\%$, etc. These sample sizes are obtained by replacing ε in Theorem 2.1 with $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$, due to Theorem 3.1, as we discussed that this is a possibility at the end of Section 3.2.

Table 2: The worst case risk as well as the recall of the generated hypotheses using FIND-S under the uniform distribution over 1,000 runs in the traditional PAC framework (Definition 2.3). Note that the recall of the generated hypotheses can be dramatically low in the traditional PAC framework.

Minority Rate (p)	**Max Risk**	**Recall**			
		Min	**Median**	**Mean**	**Max**
25.0%	0	1	1	1	1
12.5%	0	1	1	1	1
6.25%	0	1	1	1	1
3.125%	0	1	1	1	1
1.563%	0	1	1	1	1
0.781%	0.781%	$4 \cdot 10^{-10}$	1	0.886	1
0.391%	0.391%	$2 \cdot 10^{-28}$	0.25	0.389	1
0.195%	0.195%	$4 \cdot 10^{-28}$	$3 \cdot 10^{-5}$	0.078	1
0.098%	0.098%	$8 \cdot 10^{-28}$	$2 \cdot 10^{-13}$	0.001	1
0.049%	0.049%	$1 \cdot 10^{-27}$	$2 \cdot 10^{-27}$	$2 \cdot 10^{-4}$	0.063
0.024%	0.024%	$3 \cdot 10^{-27}$	$3 \cdot 10^{-27}$	$1 \cdot 10^{-5}$	0.008

As a last remark, for the case with high recall and high precision, we note that we *did obtain* some imperfect solutions for other values of the parameters. For example, this was the case when we set $\lambda = 0.99$ and $n = 500$ and moreover the target had size $k = 200$ corresponding to a minority rate $p \approx 13.4\%$. It is a question if this can be observed in practice for lower minority rates, when $\min\{\varepsilon, \gamma p_b, \xi p_b/2\}$ will not be determined by ε. For example, consider hypotheses that have all the

good variables plus some bad variables. For any such hypothesis h it holds $R_D(h,c) \leq \mathbf{Pr}_{x\sim D}(c(x)=1) = p$. Moreover, if we investigate cases where $p \leq \varepsilon$, it follows that $R_D(h,c) \leq \varepsilon$. In addition, for any such hypothesis h that has $w \leq \left\lfloor \log_{1/\lambda}(1/(1-\gamma)) \right\rfloor$ bad variables in excess to the $|c|$ good variables, by Proposition 4.1 it holds that $\text{REC}_D(h,c) \geq 1-\gamma$. Therefore, it is a question if such solutions can be returned by FIND-S (or some other empirical risk minimization process).

4.2.2 Experiments with the SWAPPING ALGORITHM In the case of the SWAPPING ALGORITHM we use similar parameters; i.e., $\varepsilon = 0.05, \delta = 0.1, \gamma = 0.4, \xi = 0.9$. The algorithm identifies the target precisely, when the target has size $|c| \leq \left\lceil \lg\left(\frac{3}{2\cdot 0.05}\right)\right\rceil = 5$, corresponding to minority rates $25\%, 12.5\%, 6.25\%$, and 3.125%. The rest of the cases for learning in the traditional PAC framework are shown in Table 3. We observe that as the rate p of the minority class decreases, so does the recall and the precision of the generated hypotheses.

In the case where we also want high precision and high recall, for target sizes $k \in \{6,7,\ldots,12\}$ corresponding to minority class rates $\{2^{-6}, 2^{-7}, \ldots, 2^{-12}\}$, we have that $\min\{\varepsilon, \gamma p_b, \xi p_b/2\} = \gamma p_b = 0.4 \cdot 2^{-k}$. Therefore, the solutions returned by the algorithm can have at most $\left\lceil \lg\left(3/(2\cdot 0.4\cdot 2^{-k})\right)\right\rceil = k+2$ variables, and as a result in every case the algorithm identifies the target, achieving risk 0, recall 1, and precision 1.

5 Conclusion

We extended the framework of PAC learning to also include high recall and high precision among its goals so that we can have a better theory for imbalanced datasets. We described how traditional PAC algorithms can be adapted to this new setting and also extended basic results on PAC learning in this new setting. We proposed a method to compute a lower bound on the true unknown constant rate p of the minority class. Eventually we studied the solutions obtained by two algorithms for learning monotone conjunctions in both PAC frameworks.

Our new framework poses traditional algorithms under new light and urges us to explore directions of designing new and intuitive algorithms that have explicitly the notions of recall and precision in mind. The need for new intuitive algorithms is sought for in other situations as well; see, e.g., [24].

References

[1] Dana Angluin and Philip D. Laird. Learning From Noisy Examples. *Machine Learning*, 2(4):343–370, 1987.

[2] Martin H.G. Anthony and Norman L. Biggs. *Computational Learning Theory: An Introduction*. Cambridge University Press, 1992.

[3] Liliya Besaleva and Alfred C. Weaver. Classification of imbalanced data in E-commerce. In *2017 Intelligent Systems Conference (IntelliSys)*, pages 744–750. IEEE, 2017.

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's Razor. *Information Processing Letters*, 24(6):377–380, 1987.

[5] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[6] Paula Branco, Luís Torgo, and Rita P. Ribeiro. A Survey of Predictive Modeling on Imbalanced Domains. *ACM Computing Surveys*, 49(2):31:1–31:50, 2016.

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1565–1576, 2019.

[8] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[9] Dimitrios I. Diochnos. On the Evolution of Monotone Conjunctions: Drilling for Best Approximations. In *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, pages 98–112, 2016.

[10] Dimitrios I. Diochnos and György Turán. On Evolvability: The Swapping Algorithm, Product Distributions, and Covariance. In *Stochastic Algorithms: Foundations and Applications, 5th International Symposium, SAGA 2009, Sapporo, Japan, October 26-28, 2009. Proceedings*, pages 74–88, 2009.

[11] Haibo He and Edwardo A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[12] Taeho Jo and Nathalie Japkowicz. Class imbalances versus small disjuncts. *SIGKDD Explorations*, 6(1):40–49, 2004.

[13] Thorsten Joachims. A support vector method for multivariate performance measures. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 377–384. ACM, 2005.

[14] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, USA, 1994.

[15] Wael Khreich, Eric Granger, Ali Miri, and Robert Sabourin. Iterative Boolean combination of classifiers

Table 3: The best-case and worst-case risk, the recall and the precision of the generated hypotheses using the SWAPPING ALGORITHM under the uniform distribution over 1,000 runs in the traditional PAC framework (Definition 2.3). Note that while the risk of the generated hypotheses satisfies the PAC criterion (i.e., less than $\epsilon = 5\%$), nevertheless, the attained values for the recall/precision on the generated hypotheses can be very low. It is precisely situations such as this one, as well as in Table 2, that we want to address with the proposed framework.

Minority	**Risk**		**Recall**				**Precision**			
Rate (p)	**Min**	**Max**	**Min**	**Median**	**Mean**	**Max**	**Min**	**Median**	**Mean**	**Max**
1.563%	1.563%	1.563%	1	1	1	1	50.0%	50.0%	50.0%	50.0%
0.781%	2.344%	3.857%	3.125%	1	70.375%	1	0.781%	25.0%	17.594%	25.0%
0.391%	2.734%	3.491%	3.125%	6.250%	33.494%	1	0.391%	0.781%	4.187%	12.5%
0.195%	2.930%	3.308%	3.125%	3.125%	9.559%	1	0.195%	0.195%	0.597%	6.250%
0.098%	3.027%	3.217%	3.125%	3.125%	5.734%	1	0.098%	0.098%	0.179%	3.125%
0.049%	3.149%	3.171%	3.125%	3.125%	5.216%	25.0%	0.049%	0.049%	0.081%	0.391%
0.024%	3.125%	3.148%	3.125%	3.125%	5.450%	50.0%	0.024%	0.024%	0.043%	0.391%

in the ROC space: An application to anomaly detection with HMMs. *Pattern Recognition*, 43(8):2732–2752, 2010.

[16] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[17] Philip D. Laird. *Learning from Good and Bad Data*. Kluwer Academic Publishers, USA, 1988.

[18] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics Part B*, 39(2):539–550, 2009.

[19] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

[20] Amy McGovern, Kimberly L. Elmore, David John Gagne, Sue Ellen Haupt, Christopher D. Karstens, Ryan Lagerquist, Travis Smith, and John K. Williams. Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, 98(10):2073 – 2090, 01 Oct. 2017.

[21] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[22] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018.

[23] Harikrishna Narasimhan, Harish G. Ramaswamy, Aadirupa Saha, and Shivani Agarwal. Consistent multiclass algorithms for complex performance measures. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2398–2407. JMLR.org, 2015.

[24] Lev Reyzin. Statistical Queries and Statistical Algorithms: Foundations and Applications. *CoRR*, abs/2004.00557, 2020.

[25] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA, 2014.

[26] Yanmin Sun, Mohamed S. Kamel, Andrew K. C. Wong, and Yang Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[27] Yanmin Sun, Andrew K. C. Wong, and Mohamed S. Kamel. Classification of Imbalanced Data: a Review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4):687–719, 2009.

[28] Theodore B. Trafalis, Indra Adrianto, Michael B. Richman, and S. Lakshmivarahan. Machine-learning classifiers for imbalanced tornado data. *Computational Management Science*, 11(4):403–418, 2014.

[29] György Turán. A Survey of Some Aspects of Computational Learning Theory (Extended Abstract). In *Fundamentals of Computation Theory, 8th International Symposium, FCT '91, Gosen, Germany, September 9-13, 1991, Proceedings*, pages 89–103, 1991.

[30] Leslie G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[31] Leslie G. Valiant. Evolvability. *Journal of the ACM*, 56(1):3:1–3:21, 2009.

[32] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971.

[33] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to Model the Tail. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 7029–7039, 2017.

[34] Gang Wu and Edward Y. Chang. Aligning Boundary in Kernel Space for Learning Imbalanced Dataset. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1-4 November 2004, Brighton, UK*, pages 265–272, 2004.