A Complete and Tight Average-Case Analysis of Learning Monomials

Rüdiger Reischuk[1], [⋆] and Thomas Zeugmann[2], [⋆⋆]

[1] Institut für Theoretische Informatik, Med. Universität zu Lübeck, Wallstraße 40, 23560 Lübeck, Germany
`reischuk@informatik.mu-luebeck.de`
[2] Department of Informatics, Kyushu University, Kasuga 816-8580, Japan
`thomas@i.kyushu-u.ac.jp`

Abstract. We advocate to analyze the average complexity of learning problems. An appropriate framework for this purpose is introduced. Based on it we consider the problem of *learning monomials* and the special case of learning *monotone* monomials *in the limit* and for *on-line predictions* in two variants: from positive data only, and from positive and negative examples. The well-known *Wholist algorithm* is completely analyzed, in particular its average-case behavior with respect to the class of *binomial distributions.* We consider different complexity measures: the *number of mind changes, the number of prediction errors, and the total learning time.* Tight bounds are obtained implying that worst case bounds are too pessimistic. On the average learning can be achieved *exponentially faster.*

Furthermore, we study a new learning model, *stochastic finite learning*, in which, in contrast to PAC learning, some information about the underlying distribution is given and the goal is to find a *correct* (not only approximatively correct) hypothesis. We develop techniques to obtain good bounds for stochastic finite learning from a precise average case analysis of strategies for learning in the limit and illustrate our approach for the case of learning monomials.

1. Introduction

Learning concepts efficiently has attracted considerable attention during the last decade. However, research following the traditional lines of inductive inference has mainly considered the update time, i.e., the effort to compute a *single* new hypothesis. Starting with Valiant's paper [18], the total amount of time needed to solve a given learning problem has been investigated as well. The

⋆ Part of this work was performed while visiting the Department of Informatics at Kyushu University supported by the Japan Society for the Promotion of Science under Grant JSPS 29716102.

⋆⋆ Supported by the Grant-in-Aid for Scientific Research in Fundamental Areas from the Japanese Ministry of Education, Science, Sports, and Culture under grant no. 10558047.

complexity bounds proved within the PAC model are usually *worst-case* bounds. In experimental studies large gaps have often been observed between the time bounds obtained by a mathematical analysis and the actual runtime of a learner on typical data. This phenomenon can be explained easily. Data from running tests provide information about the *average-case* performance of a learner, rather than its worst-case behavior. Since algorithmic learning has a lot of practical applications it is of great interest to analyze the average-case performance, and to obtain tight bounds saying something about the typical behavior in practice.

Pazzani and Sarrett [14] have proposed a framework for analyzing the average-case behavior of learning algorithms. Several authors have followed their approach (cf., e.g., [12, 13]). Their main goal is to predict the expected accuracy of the hypothesis produced with respect to the number of training examples. However, the results obtained so far are not satisfactory. Typically, the probability that a random example is misclassified by the current hypothesis is estimated by a complicated formula. The *evaluation* of this formula, and the computation of the corresponding expectation has been done by Monte-Carlo simulations. Clearly, such an approach does not provide general results about the average-case behavior for broader classes of distributions. Moreover, it is hard to compare these bounds with those proved for the PAC model.

We outline a new setting to study the average-case behavior of learning algorithms overcoming these drawbacks and illustrate it for learning monomials.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \dots\}$ be the set of all natural numbers, and let $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$. If M is a set, $|M|$ is used for its cardinality. For an infinite sequence d and $j \in \mathbb{N}^+$ let $\boldsymbol{d[j]}$ denote the initial segment of d of length j. By $(0,1)$ we denote the real interval from 0 to 1 excluding both endpoints. For $n \in \mathbb{N}^+$, let $\mathcal{X}_n = \{0,1\}^n$ be the ***learning domain*** and $\wp(\mathcal{X}_n)$ the power set of $\mathcal{X}_n$. A subset c of $\mathcal{X}_n$ is called a ***concept***, and a subset $\mathcal{C}$ of $\wp(\mathcal{X}_n)$ a ***concept class***. The notation c is also used to denote the characteristic function of a subset, that is for $b \in \mathcal{X}_n$: $c(b) = 1$ iff $b \in c$. To define the classes of concepts we deal with in this paper let $\mathcal{L}_n = \{x_1, \bar{x}_1, x_2, \bar{x}_2 \dots, x_n, \bar{x}_n\}$ be a set of literals. x_i is a *positive* literal and $\bar{x}_i$ a *negative* one. A conjunction of literals defines a ***monomial***. For a monomial m let $\boldsymbol{\#(m)}$ denote its length, that is the number of literals in it.

m describes a subset $L(m)$ of $\mathcal{X}_n$, in other words a concept, in the obvious way: the concept contains exactly those binary vectors for which the monomial evaluates to 1, that is $L(m) := \{b \in \mathcal{X}_n \mid m(b) = 1\}$. The collection of objects we are going to learn is the set $\boldsymbol{\mathcal{C}_n}$ of all concepts that are describable by monomials over $\mathcal{X}_n$. There are two trivial concepts, the empty subset and $\mathcal{X}_n$ itself. $\mathcal{X}_n$, which will also be called "TRUE", can be represented by the empty monomial. The concept "FALSE" has several descriptions. To avoid ambiguity, we always represent "FALSE" by the monomial $x_1\bar{x}_1 \dots x_n\bar{x}_n$. Furthermore, we often identify the set of all monomials over $\mathcal{L}_n$ and the concept class $\mathcal{C}_n$. Note that $|\mathcal{C}_n| = 3^n + 1$. We also consider the subclass $\boldsymbol{\mathcal{MC}_n}$ of $\mathcal{C}_n$ consisting of those concepts that can be described by ***monotone*** monomials, i.e., by monomials containing positive literals only. It holds $|\mathcal{MC}_n| = 2^n$.

3. Learning Models and Complexity Measures

The first learning model we are dealing with is the ***on-line prediction*** model going back to Barzdin, Freivald [1] and Littlestone [10]. In this setting the source of information is specified as follows. The learner is given a sequence of labeled examples $d = \langle d_j \rangle_{j \in \mathbb{N}^+} = \langle b_1,\, c(b_1),\, b_2,\, c(b_2),\, b_3,\, c(b_3),\, \ldots \rangle$ from the concept c, where the $b_j \in \mathcal{X}_n$, and $c(b_j) = 1$ if $b_j \in c$ and $c(b_j) = 0$ otherwise. The examples b_j are picked arbitrarily and the information provided is assumed to be without any errors. We refer to such sequences as ***data sequences*** and use $data(c)$ to denote the set of all data sequences for concept c.

A learner $\boldsymbol{P}$ must predict $c(b_j)$ after having seen $d[2j-1] = \langle b_1,\, c(b_1), \ldots,$ $b_{j-1},\, c(b_{j-1}), b_j \rangle$. We denote this hypothesis by $P(d[2j-1])$. Then it receives the true value $c(b_j)$ and the next Boolean vector b_{j+1}. The learner has successfully learned if it eventually reaches a point beyond which it always predicts correctly.

Definition 1. *A concept class $\mathcal{C}$ is called* ***on-line predictable*** *if there is a learner P such that for all concepts $c \in \mathcal{C}$ and all data sequences $d = \langle d_j \rangle_{j \in \mathbb{N}^+} \in data(c)$ it holds: $P(d[2j-1])$ is defined for all j, and $P(d[2j-1]) = d_{2j}$ for all but finitely many j.*

For on-line prediction, the ***complexity measure*** considered is the *number of prediction errors* made. Note that the prediction goal can always be achieved trivially if the learning domain is finite. Therefore, we aim to minimize the number of prediction errors when learning monomials.

Next, let us define Gold-style [5] ***learning in the limit***. One distinguishes between learning from positive and negative data, and learning from positive data only. For a concept c, let $info(c)$ be the set of those data sequences $\langle b_1,\, c(b_1),\, b_2,\, c(b_2),\, b_3,\, c(b_3),\, \ldots \rangle$ in $data(c)$ that contain each element b of the learning domain $\mathcal{X}_n$ at least once. Such a sequence is called ***informant***. For ease of notation let us pair each element b_j with its classification $c(b_j)$. Then the j-th entry of an informant sequence d will be $d_j := (b_j, c(b_j))$.

A ***positive presentation*** of c is a data sequence that contains only elements of c and each one at least once. Thus all the values $c(b_j)$ are equal to 1 and thus could be omitted. In this case we will denote the sequence simply by $d = \langle d_j \rangle_{j \in \mathbb{N}^+} = \langle b_1, b_2, b_3, \ldots \rangle$. Let $d[j]^+ := \{ b_i \mid 1 \le i \le j \}$ be the set of all examples contained in the prefix of d of length j, and let $pos(c)$ denote the set of all positive presentations of c. The elements of $pos(c)$ are also called a ***text*** for c.

A limit learner is an ***inductive inference machine*** (abbr. IIM). An IIM M works as follows. As inputs it gets incrementally growing segments of a positive presentation (resp. of an informant) d. After each new input, it outputs a hypothesis $M(d[j])$ from a predefined hypothesis space $\mathcal{H}$. Each hypothesis refers to a unique element of the concept class.

Definition 2. *Let $\mathcal{C}$ be a concept class and let $\mathcal{H}$ be a hypothesis space for it. $\mathcal{C}$ is called* ***learnable in the limit*** *from positive presentation (resp. from informant) if there is an IIM M such that for every $c \in \mathcal{C}$ and every $d \in pos(c)$ (resp. $d \in info(c)$): $M(d[j])$ is defined for all j, and $M(d[j]) = h$ for all but finitely many j, where $h \in \mathcal{H}$ is a hypothesis referring to c.*

For the concept class $\mathcal{C}_n$ we choose as hypothesis space the set of all monomials over $\mathcal{L}_n$, whereas for $\mathcal{MC}_n$ it is the set of all monotone monomials. Again, we are interested how efficiently $\mathcal{C}_n$ and $\mathcal{MC}_n$ can be learned in the limit.

The first complexity measure we consider is the ***mind change*** complexity. A mind change occurs iff $M(d[j]) \neq M(d[j+1])$. Clearly, this measure is closely related to the number of prediction errors. Both complexity measures say little about the total amount of data and time needed until a concept is guessed correctly. Thus, for learning in the limit we also measure the *time complexity.* As in [3] we define the total learning time as follows. Let M be any IIM learning a concept class $\mathcal{C}$ in the limit. Then, for $c \in \mathcal{C}$ and a text or informant d for c, let $\boldsymbol{Con(M,d)}$ = the least $i \in \mathbb{N}^+$ such that $M(d[j]) = M(d[i])$ for all $j \geq i$ denote the ***stage of convergence*** of M on d (cf. [5]). Moreover, by $T_M(d_j)$ we denote the number of steps to compute $M(d[j])$. We measure this quantity as a function of the length of the input and refer to it as the *update time.* Finally, the ***total learning time*** taken by the IIM M on a sequence d is defined as $\boldsymbol{TT(M,d)} := \sum_{j=1}^{Con(M,d)} T_M(d[j])$. Given a probability distribution D on the data sequences d we evaluate the *expectation* of $TT(M,d)$ with respect to D, the ***average total learning time***.

4. The Wholist Algorithm Learning Monomials

Next, we present Haussler's [6] Wholist algorithm for on-line prediction of monomials. For learning in the limit this algorithm can be modified straightforwardly. The limit learner computes a new hypothesis using only the most recent example received and his old hypothesis. Such learners are called *iterative* (cf. [8]). Let $c \in \mathcal{C}_n$, let $d \in data(c)$, and let $b_i = b_i^1 b_i^2 \ldots b_i^n$ denote the i-th Boolean vector in d. Recall that "TRUE" is represented by the empty monomial.

***Algorithm $\mathcal{P}$*:** On input sequence $\langle b_1,\, c(b_1),\, b_2,\, c(b_2),\, \ldots\rangle$ do the following:

```
Initialize  $h_0 := x_1\bar{x}_1 \ldots x_n\bar{x}_n$.
for $i = 1, 2, \ldots$ do
    let $h_{i-1}$ denote $\mathcal{P}$'s internal hypothesis produced before receiving $b_i$;
    when receiving $b_i$ predict $h_{i-1}(b_i)$; read $c(b_i)$;
    if $h_{i-1}(b_i) = c(b_i)$ then $h_i := h_{i-1}$
        else for $j := 1$ to $n$ do
            if $b_i^j = 1$ then delete $\bar{x}_j$ in $h_{i-1}$ else delete $x_j$ in $h_{i-1}$;
        let $h_i$ be the resulting monomial
end.
```

Note that the algorithm is monotone with respect to the sequence of its internal hypotheses: $h_i \geq h_{i-1}$ when considered as function on $\mathcal{X}_n$.

Theorem 1. *Algorithm $\mathcal{P}$ learns the set of all monomials within the prediction model. It makes at most $n+1$ prediction errors.*

To learn the monotone concept class $\mathcal{MC}_n$, algorithm $\mathcal{P}$ can be easily modified by initializing $h_0 = x_1 x_2 \ldots x_n$ and by simplifying the loop appropriately. We refer to the modified algorithm as to $\mathcal{MP}$.

Theorem 1 can be directly reproved for $\mathcal{MP}$ with the only difference that now the worst-case bound for the number of prediction errors is n instead of $n+1$.

5. Complexity Analysis: Best and Worst Case

For the learning models defined above we estimate the best-case complexity, the worst-case complexity, and the expectation of algorithm $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{MP}}$. We start with the first two issues. Both algorithms do not make any prediction errors iff the initial hypothesis h_0 equals the target monomial. For $\boldsymbol{\mathcal{P}}$ this means that the concept to be learned is "FALSE", while for $\boldsymbol{\mathcal{MP}}$ the concept is the all-1 vector. These special concepts can be considered as ***minimal*** in their class. For them the best-case and the worst-case number of predictions errors coincide.

In the general case, we call the literals in a monomial m ***relevant***. All other literals in $\mathcal{L}_n$ (resp. in $\{x_1, \ldots, x_n\}$ in the monotone case) are said to be ***irrelevant*** for m. There are $2n - \#(m)$ irrelevant literals in general, and $n - \#(m)$ in the monotone case. We call bit i **relevant** for m if x_i or $\bar{x}_i$ is relevant for m. By $\boldsymbol{k} = \boldsymbol{k(m)} = n - \#(m)$ we denote the number of irrelevant bits.

Theorem 2. *Let $c = L(m)$ be a non-minimal concept in $\mathcal{MC}_n$. Then algorithm $\boldsymbol{\mathcal{MP}}$ makes 1 prediction error in the best case, and $k(m)$ prediction errors in the worst-case.*
If c is a non-minimal concept of $\mathcal{C}_n$ algorithm $\boldsymbol{\mathcal{P}}$ makes 2 prediction errors in the best case and $1 + k(m)$ prediction errors in the worst-case.

As Theorem 2 shows, the gap between the best-case and worst-case behavior can be quite large. Thus, we ask what are the expected bounds for the number of prediction errors on randomly generated data sequences. Before answering this question we estimate the worst-case number of prediction errors averaged over the whole concept class $\mathcal{MC}_n$, resp. $\mathcal{C}_n$. Thus we get a complexity bound with respect to the parameter n, instead of $\#(m)$ as in Theorem 2. This averaging depends on the underlying probability distribution for selecting the target concepts (for the corresponding data sequences we consider the worst input). The average is shown to be linear in n if the literals are binomially distributed.

To generate the probability distributions we assume for $\mathcal{MC}_n$ the relevant positive literals to be drawn independently at random with probability p, $p \in (0, 1)$. Thus, with probability $1-p$ a literal is irrelevant. The length of the monomials drawn by this distribution is binomially distributed with parameter p. Thus we call such a distribution on the concept class a ***binomial distribution***.

Theorem 3. *Let the concepts in $\mathcal{MC}_n$ be binomially distributed with parameter p. Then the average number of prediction errors of $\boldsymbol{\mathcal{MP}}$ for the worst data sequences is $n(1 - p)$.*

In case $p = 1/2$, the bound says that the maximal number of prediction errors when uniformly averaged over all concepts in $\mathcal{MC}_n$ is $n/2$.

Next, we deal with the class $\mathcal{C}_n$. For comparing it to the monotone case we have to clarify what does it mean for concepts in $\mathcal{C}_n$ to be binomially distributed. Since there are $3^n + 1$ many concepts in $\mathcal{C}_n$ for a uniform distribution each concept must have probability $1/(3^n + 1)$. For each position $i = 1, \ldots, n$ three options are possible, i.e., we may choose x_i, $\bar{x}_i$ or neither of them. This suggests the formula $\binom{n}{k_1, k_2, k_3} p_1^{k_1} p_2^{k_2} p_3^{k_3}$, where p_1 is the probability to take x_i, p_2 the probability to choose $\bar{x}_i$ and p_3 the probability to choose none, and $p_1 + p_2 + p_3 = 1$

and $k_1 + k_2 + k_3 = n$. $k_1 + k_2$ counts the number of relevant literals, resp. bits. However, this formula does not include the concept "FALSE." Thus, let us introduce $p_f \in (0,1)$ for the probability to choose "FALSE." Then the formula becomes $(1-p_f)\binom{n}{k_1,k_2,k_3} p_1^{k_1} p_2^{k_2} p_3^{k_3}$. We call such a probability distribution a ***weighted multinomial distribution*** with parameters (p_f, p_1, p_2, p_3).

Theorem 4. *Let the concepts in $\mathcal{C}_n$ occur according to a weighted multinomial distribution with parameters (p_f, p_1, p_2, p_3). Then the average number of prediction errors of $\boldsymbol{\mathcal{P}}$ for the worst data sequences is $(1-p_f)(1+np_3)$.*

For the particular case that all concepts from $\mathcal{C}_n$ are equally likely, i.e., $p_1 = p_2 = p_3 = 1/3$ and $p_f = 1/(3^n + 1)$, we directly get that on the average less than $n/3 + 1$ errors are to be expected given the worst data sequences. Hence, in this case the class $\mathcal{C}_n$ seems to be easier to learn than $\mathcal{MC}_n$ with respect to the complexity measure *prediction errors*. However, this impression is a bit misleading, since the probabilities to generate an irrelevant literal are *different*, i.e., $1/3$ for $\mathcal{C}_n$ and $1/2$ for $\mathcal{MC}_n$. If we assume the probabilities to generate an irrelevant literal to be equal, say q, and make the meaningful assumption that "FALSE" has the same probability as "TRUE" then the average complexity is $\frac{1}{1+q^n}(1+nq)$ for $\mathcal{C}_n$ and nq for $\mathcal{MC}_n$. Since for $\mathcal{C}_n$ it holds $q = 1-(p_1+p_2)$, and for $\mathcal{MC}_n$ $q = 1-p$, under these assumptions $\mathcal{MC}_n$ is easier to learn than $\mathcal{C}_n$. This insight is interesting, since it clearly shows the influence of the underlying distribution. In contrast, previous work has expressed these bounds in terms of the VC-dimension which is the same for both classes, i.e., n.

The results above directly translate to learning in the limit from informant or from positive presentations for the complexity measure *number of mind changes.*

What can be said about the total learning time? The best-case can be handled as above. Sine the update time is linear in n for both algorithms $\boldsymbol{\mathcal{MP}}$ and $\boldsymbol{\mathcal{P}}$, in the best case the total learning time is linear. The worst-case total learning time is *unbounded* for both algorithms, since every text and informant may contain as many repetitions of data not possessing enough information to learn the target.

Hence, as far as learning in the limit and the complexity measure *total learning time* are concerned, there is a huge gap between the best-case and the worst-case behavior. Since the worst-case is unbounded, it does not make sense to ask for an analogue to Theorem 3 and 4. Instead, we continue by studying the average-case behavior of the limit learner $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{MP}}$.

6. Average-Case Analysis for Learning in the Limit from Text

For the following average case analysis we assume that the data sequences are generated at random with respect to some probability distribution D taken from a class of admissible distributions $\mathcal{D}$ specified below. We are interested in the *average number* of examples till an algorithm has converged to a correct hypothesis. **CON** denotes a random variable counting the number of examples till convergence. Let d be a text of the concept c to be learned that is generated at random according to D. If the concept to be learned is "FALSE" no examples are needed. Otherwise, if the target concept contains precisely n literals then one positive example suffices (note that this one is unique). Thus, for these two cases everything is clear and the probability distributions D on the set of positive examples for c are trivial.

For analyzing the nontrivial cases, let $c = L(m) \in \mathcal{C}_n$ be a concept with monomial $m = \bigwedge_{j=1}^{\#(m)} \ell_{i_j}$ such that $k = k(m) = n - \#(m) > 0$. There are 2^k positive examples for c. For the sake of presentation, we assume these examples to be ***binomially distributed***. That is, in a random positive example all entries corresponding to irrelevant bits are selected independently of each other. With some probability p this will be a 1, and with probability $q := 1-p$ a 0. We shall consider only nontrivial distributions where $0 < p < 1$. Note that otherwise the data sequence does not contain all positive examples. We aim to compute the expected number of examples taken by $\boldsymbol{\mathcal{P}}$ until convergence.

The first example received forces $\boldsymbol{\mathcal{P}}$ to delete precisely n of the $2n$ literals in h_0. Thus, this example always plays a *special* role. Note that the resulting hypothesis h_1 depends on b_1, but the number k of literals that remain to be deleted from h_1 until convergence is *independent* of b_1. Using tail bound techniques, we can show the following theorem.

Theorem 5. *Let $c = L(m)$ be a non-minimal concept in $\mathcal{C}_n$, and let the positive examples for c be binomially distributed with parameter p. Define $\psi := \min\{\frac{1}{1-p}, \frac{1}{p}\}$. Then the expected number of positive examples needed by algorithm $\boldsymbol{\mathcal{P}}$ until convergence can be bounded by $E[\mathrm{CON}] \leq \lceil \log_\psi k(m) \rceil + 3$.*

A similar analysis can be given in the monotone setting for algorithm $\boldsymbol{\mathcal{MP}}$.

Corollary 6. *For every binomially distributed text with parameter $0 < p < 1$ the average total learning time of algorithm $\boldsymbol{\mathcal{P}}$ for concepts in $\mathcal{C}_n$ with μ literals is at most $O(n(\log(n - \mu + 2))$.*

The expectation alone does not provide complete information about the average case behavior of an algorithm. We also like to deduce bounds on how often the algorithm exceeds the average considerably. The Wholist algorithm possesses two favorable properties that simplify this derivation considerably, i.e., it is ***set-driven*** and ***conservative***. Set-driven means that for all $c \in \mathcal{C}_n$ all $d,\ h \in pos(c)$ and all $i, j \in \mathbb{N}^+$ the equality $d[i]^+ = h[j]^+$ implies $\boldsymbol{\mathcal{P}}(d[i]) = \boldsymbol{\mathcal{P}}(h[j])$. A learner is said to be conservative if every mind change is caused by an inconsistency with the data seen so far. Clearly, the Wholist algorithm satisfies this condition, too. Now, the following theorem establishes exponentially shrinking tail bounds for the expected number of examples needed in order to achieve convergence.

Theorem 7 ([16]). *Let CON be the sample complexity of a conservative and set-driven learning algorithm. Then $\Pr[\mathrm{CON} > 2t \cdot E[\mathrm{CON}]] \leq 2^{-t}$ for all $t \in \mathbb{N}$.*

A simple calculation shows that in case of exponentially shrinking tail bounds the variance is bounded by $O(E[\mathrm{CON}]^2)$.

7. Stochastic Finite Learning

Next we shall show how to convert the Wholist algorithm into a text learner that identifies all concepts in $\mathcal{C}_n$ *stochastically in a bounded number of rounds with high confidence*. A bit *additional knowledge* concerning the underlying class of probability distributions is required. Thus, in contrast to the PAC model, the resulting learning model is *not* distribution-free. But with respect to the quality of its hypotheses, it is stronger than the PAC model by requiring the output to be *probably exactly correct* rather than *probably approximately correct.* The *main* advantage is the usage of the additional knowledge to reduce the sample size, and hence the total learning time drastically. This contrasts to previous work in the area of PAC learning (cf., e.g., [2, 4, 7, 9, 11, 17]). These papers have shown concepts classes to be PAC learnable from polynomially many examples given a known distribution or class of distributions, while the general PAC learnability of these concepts classes is not achievable or remains open. Note that our general approach, i.e., performing an average-case analysis and proving exponentially shrinking tail bounds for the expected total learning time, can also be applied to obtain results along this line (cf. [15, 16]).

Definition 3. *Let $\mathcal{D}$ be a set of probability distributions on the learning domain, $\mathcal{C}$ a concept class, $\mathcal{H}$ a hypothesis space for $\mathcal{C}$, and $\delta \in (0,1)$. $(\mathcal{C},\mathcal{D})$ is said to be* ***stochastically finite learnable with δ-confidence****with respect to $\mathcal{H}$ iff there is an IIM M that for every $c \in \mathcal{C}$ and every $D \in \mathcal{D}$ performs as follows. Given a random presentation d for c generated according to D, M stops after having seen a finite number of examples and outputs a single hypothesis $h \in \mathcal{H}$. With probability at least $1-\delta$ (with respect to distribution D) h has to be correct, that is $L(h) = c$ in case of monomials. If stochastic finite learning can be achieved with δ-confidence for every $\delta > 0$ then we say that $(\mathcal{C},\mathcal{D})$ can be learned stochastically finite* ***with high confidence****.*

We study the case that the positive examples are binomially distributed with parameter p. But we do not require precise knowledge about the underlying distribution. Instead, we reasonably assume that *prior knowledge* is provided by parameters p_{low} and p_{up} such that $p_{low} \le p \le p_{up}$ for the true parameter p. Binomial distributions fulfilling this requirement are called ***(p_{low}, p_{up})–admissible distributions***. Let $\boldsymbol{\mathcal{D}_n[p_{low}, p_{up}]}$ denote the set of such distributions on $\mathcal{X}_n$.

If bounds p_{low} and p_{up} are available, the Wholist algorithm can be transformed into a stochastic finite learner inferring all concepts with high confidence.

Theorem 8. *Let $0 < p_{low} \le p_{up} < 1$ and $\psi := \min\{\frac{1}{1-p_{low}}, \frac{1}{p_{up}}\}$. Then $(\mathcal{C}_n, \mathcal{D}_n[p_{low}, p_{up}])$ is stochastically finitely learnable with high confidence from positive presentations. To achieve δ-confidence no more than $O(\log_2 1/\delta \cdot \log_\psi n)$ many examples are necessary.*

The latter example bound can even be improved to $\log_\psi n + \log_\psi 1/\delta + O(1)$ by performing a careful *error analysis*, i.e., for the Wholist algorithm, the confidence requirement increases the sample size by an additive term $\log_\psi 1/\delta$ only.

8. Average-case Analysis for Learning in the Limit from Informant

Finally, we consider how the results obtained so far translate to the case of learning from informant. First, we investigate the uniform distribution over $\mathcal{X}_n$. Again, we have the trivial cases that the target is "FALSE" or m is a monomial without irrelevant bits. In the first case, no example is needed at all, while in the latter one, there is only one positive example having probability 2^{-n}. Thus the expected number of examples needed until successful learning is $2^n = 2^{\#(m)}$.

Theorem 9. *Let $c = L(m) \in \mathcal{C}_n$ be a nontrivial concept. If an informant for c is generated from the uniform distribution by independent draws the expected number of examples needed by algorithm $\boldsymbol{\mathcal{P}}$ until convergence is bounded by* $E[\mathrm{CON}] \ \leq\ 2^{\#(m)}\ (\lceil \log_2 k(m) \rceil + 3)$.

Hence, as long as $k(m) = n - O(1)$, we still achieve an expected total learning time $O(n \log n)$. But if $\#(m) = \Omega(n)$ the expected total learning is exponential. However, if there are many relevant literals then even h_0 may be considered as a not too bad *approximation* for c. Thus, let $\varepsilon \in (0,1)$ be an error parameter as in the PAC model. We ask if one can achieve an expected sample complexity for computing an ε-approximation that is polynomially bounded in $\log n$ and $1/\varepsilon$.

Let $err_m(h_j) := D(L(h_j) \triangle L(m))$ be the error made by hypothesis h_j with respect to monomial m. Here $L(h_j) \triangle L(m)$ is the symmetric difference of $L(h_j)$ and $L(m)$, and D the probability distribution with respect to which the examples are drawn. h_j is an $\boldsymbol{\varepsilon}$***–approximation*** for m if $err_m(h_j) \leq \varepsilon$. Finally, we redefine the stage of convergence. Let $d = (d_j)_{j \in \mathbb{N}^+} \in \mathit{info}(L(m))$, then

$\mathbf{CON}_{\boldsymbol{\varepsilon}}\boldsymbol{(d)}$ = the least number j such that $err_m(P(d[i])) \leq \varepsilon$ for all $i \geq j$.

Note that once the Wholist algorithm has reached an ε-approximate hypothesis all further hypotheses will also be at least that close to the target monomial. The following theorem gives an affirmative answer to the question posed above.

Theorem 10. *Let $c = L(m) \in \mathcal{C}_n$ be a nontrivial concept. Assuming that examples are drawn independently from the uniform distribution, the expected number of examples needed by algorithm $\boldsymbol{\mathcal{P}}$ until converging to an ε-approximation for c can be bounded by* $E[\mathrm{CON}_\varepsilon] \ \leq\ \frac{1}{\varepsilon} \cdot (\lceil \log_2 k(m) \rceil + 3)$.

Thus, additional knowledge concerning the underlying probability distribution pays off again. Using Theorem 7 and modifying Section 7 *mutatis mutandis*, we achieve stochastic finite learning with high confidence for all concepts in $\mathcal{C}_n$ using $O(\frac{1}{\varepsilon} \cdot \log \frac{1}{\delta} \cdot \log n)$ many examples. However, the resulting learner now infers ε-approximations. Comparing this bound with the sample complexity given in the PAC model one notes an exponential reduction.

Finally, we generalize the last results to the case that the data sequences are binomially distributed for some parameter $p \in (0,1)$. This means that any particular vector containing ν times a 1 and $n - \nu$ a 0 has probability $p^\nu (1-p)^{n-\nu}$ since a 1 is drawn with probability p and a 0 with probability $1-p$. First, Theorem 9 generalizes as follows.

Theorem 11. *Let $c = L(m) \in \mathcal{C}_n$ be a nontrivial concept. Let m contain precisely π positive literals and τ negative literals. If the labeled examples for c are independently binomially distributed with parameter p and $\psi := \min\{\frac{1}{1-p},\ \frac{1}{p}\}$ then the expected number of examples needed by algorithm $\boldsymbol{\mathcal{P}}$ until convergence can be bounded by* $E[\mathrm{CON}] \ \leq\ \frac{1}{p^\pi (1-p)^\tau} \Big(\lceil \log_\psi k(m) \rceil + 3 \Big)$.

Theorem 10 directly translates into the setting of binomially distributed inputs.

Theorem 12. *Let $c = L(m) \in \mathcal{C}_n$ be a nontrivial concept. Assume that the examples are drawn with respect to a binomial distribution with parameter p, and let $\psi = \min\{\frac{1}{1-p},\ \frac{1}{p}\}$. Then the expected number of examples needed by algorithm $\boldsymbol{\mathcal{P}}$ until converging to an ε-approximation for c can be bounded by* $E[\mathrm{CON}] \ \leq\ \frac{1}{\varepsilon} \cdot (\lceil \log_\psi k(m) \rceil + 3)$.

Finally, one can also learn ε-approximations stochastically finite with high confidence from informant with an exponentially smaller sample complexity.

Theorem 13. *Let* $0 < p_{low} \leq p_{up} < 1$ *and* $\psi := \min\{\frac{1}{1-p_{low}}, \frac{1}{p_{up}}\}$. *For* $(\mathcal{C}_n, \mathcal{D}_n[p_{low}, p_{up}])$ ε*-approximations are stochastically finitely learnable with* δ*-confidence from informant for all* $\varepsilon, \delta \in (0,1)$. *Further,* $O\left(\frac{1}{\varepsilon} \cdot \log_2 1/\delta \cdot \log_\psi n\right)$, *resp.* $O\left(\frac{1}{\varepsilon} \cdot (\log_\psi 1/\delta + \log_\psi n)\right)$ *many examples suffice for this purpose.*

References

1. J.M. Barzdin, R.V. Freivald, On the prediction of general recursive functions. *Soviet Math. Doklady* 13:1224-1228, 1972.
2. G. Benedek and A. Itai. Learnability by fixed distributions. "Proc. 1988 Workshop on Computational Learning Theory," 81–90, Morgan Kaufmann, 1988.
3. R. Daley and C.H. Smith. On the complexity of inductive inference. *Inform. Control*, 69:12–40, 1986.
4. F. Denis and R. Gilleron. PAC learning under helpful distributions. "Proc. 8th International Workshop on Algorithmic Learning Theory," LNAI Vol. 1316, 132–145, Springer-Verlag, 1997.
5. E.M. Gold, Language identification in the limit. *Inform. Control* 10:447–474, 1967.
6. D. Haussler. Bias, version spaces and Valiant's learning framework. "Proc. 8th National Conference on Artificial Intelligence, 564–569, Morgan Kaufmann, 1987.
7. M. Kearns, M. Li, L. Pitt and L.G. Valiant. On the learnability of Boolean formula. "Proc. 19th Annual ACM Symposium on Theory of Computing," 285–295, ACM Press 1987.
8. S. Lange and T. Zeugmann. Incremental learning from positive data. *J. Comput. System Sci.* 53(1):88–103, 1996.
9. M. Li and P. Vitanyi. Learning simple concepts under simple distributions. *SIAM J. Comput.*, 20(5):911-935, 1991.
10. N. Littlestone. Learning quickly when irrelevant attributes are abound: A new linear threshold algorithm. *Machine Learning* 2:285–318, 1988.
11. B. Natarajan. On learning Boolean formula. "Proc. 19th Annual ACM Symposium on Theory of Computing," 295–304, ACM Press, 1987.
12. S. Okamoto and K. Satoh. An average-case analysis of k-nearest neighbor classifier. "Proc. 1st International Conference on Case-Based Reasoning Research and Development," LNCS Vol. 1010, 253–264, Springer-Verlag, 1995.
13. S. Okamoto and N. Yugami. Theoretical analysis of the nearest neighbor classifier in noisy domains. "Proc. 13th International Conference on Machine Learning, 355–363, Morgan Kaufmann 1996.
14. M.J. Pazzani and W. Sarrett, A framework for average case analysis of conjunctive learning algorithms. *Machine Learning* 9:349–372, 1992.
15. R. Reischuk and T. Zeugmann. Learning one-variable pattern languages in linear average time. "Proc. 11th Annual Conference on Computational Learning Theory," 198–208, ACM Press, 1998.
16. P. Rossmanith and T. Zeugmann. Learning k-variable pattern languages efficiently stochastically finite on average from positive data. "Proc. 4th International Colloquium on Grammatical Inference," LNAI Vol. 1433, 13–24, Springer-Verlag, 1998.
17. Y. Sakai, E. Takimoto and A. Maruoka. Proper learning algorithms for functions of k terms under smooth distributions. "Proc. 8th Annual ACM Conference on Computational Learning Theory," 206–213, ACM Press, 1995.
18. L.G. Valiant. A theory of the learnable. *Commun. ACM* 27:1134-1142, 1984.