A Foundation of Algorithmic Teaching

Shinohara, Ayumi
Department of Information Systems, Kyushu University

Miyano, Satoru
Department of Information Systems, Kyushu University

出版情報：RIFIS Technical Report. 22, 1989-12-31. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-22

RIFIS Technical Report

A Foundation of Algorithmic Teaching

Ayumi Shinohara
Satoru Miyano

December 31, 1989

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

E-mail: ayumi@rifis.sci.kyushu-u.ac.jp Phone: 092(641)1101 Ext.4458

A Foundation of Algorithmic Teaching

Ayumi Shinohara
ayumi@rifis.sci.kyushu-u.ac.jp

Satoru Miyano
miyano@rifis.sci.kyushu-u.ac.jp

Department of Information Systems
Kyushu University 39

December 31, 1989

"There is nothing so good for learning as teaching."

from The Random House Dictionary of the English Language

Abstract

We give a framework for learning from the point of teaching. In this framework we prove under some natural conditions a fact corresponding to an observation that if an adviser teaches a learner which examples to process in the order of difficulty then the learner will be able to solve a problem within given time. We also introduce a notion of teaching with a manual.

1. Introduction

The learning theory from the computational point of view has raised new notions of learnability such as "probably approximately correctly (PAC) learn" [Val84] and "minimal adequate teacher (MAT)" [Ang88]. These notions have been extensively studied from various points and many contributions are made. Good surveys are given in [Lai89] and [Pit89].

A central issue in PAC identification is to show that a concept is learnable or not. In the framework of MAT, a learner reaches a solution by making interactions with a teacher. Most discussions are made about how the learner can get necessary knowledge from the unkind teacher. But less attention has been paid to computational learning from the point of teaching. There is also a criticism that PAC identification does not seem to capture the notion of learning [Lai89].

The purpose of this paper is to provide a series of definitions about a learner which can change itself based on externally obtained information. With this framework we can consider, from the side of teaching, the problem of what to teach the learner for better achievement.

Assume that a learner tries to learn how to solve a problem through processing examples. There may be the following educational observation: If an adviser teaches the learner which examples to process in the order of difficulty, the learner will be able to solve the problem within given time.

Our main result in this paper is, roughly speaking, to show with our framework that the above observation is true under some natural conditions. We also define a notion of teaching with a manual in relation to learning.

This paper is organized as follows: Section 2 formulates a problem we deal with in this paper. In Section 3 we define a learner using oracle Turing machines and present related notions. Section 4 establishes a relationship between learnability and sample size. The main result is proved in Section 5. Discussions on teaching with a manual are given in Section 6.

2. Preliminary

Most of the papers concerned with the computational learning theory have dealt with concept learning, where the problem is to identify sets. On the other hand, [NT88], [Nat89] considered the following problem that requires solution sequences for instances.

Definition 1. Let Σ be a finite alphabet. A *problem* is a pair $D = (G, O)$, where G and O are given as follows:

1. $G : \Sigma^* \to \{0,1\}$ is a polynomial time computable function called a *goal function.*

2. O is a set $\{o_1, ..., o_k\}$, where each $o_i : \Sigma^* \to \Sigma^*$ is a polynomial time computable function of *operation.*

Let $D = (G, O)$ be a problem. For an *instance* $x \in \Sigma^*$, if there is a sequence of operations $\sigma = o_{i_t}...o_{i_2}o_{i_1}$ $(o_{i_j} \in O)$ such that $G(\sigma(x)) = 1$, we say that x is *solvable* and σ is a *solution* of x. A shortest solution of x is called *optimal.* We

denote $S(D) = \{x \in \Sigma^* \mid x \text{ is solvable}\}$. An *example* for D is a pair $\langle x, \sigma \rangle$, where σ is a solution of $x \in S(D)$. A sequence of examples $\langle x_1, \sigma_1 \rangle, ..., \langle x_m, \sigma_m \rangle$ is called a *sample* of size m for $(x_1, ..., x_m)$. An *instance distribution* or simply a *distribution* P for D is a probability distribution on $S(D)$.

3. Learner

This section gives a definition of a learner based on the following view:

> "*A learning machine is just a machine that changes itself, perhaps in part on the basis of externally obtained information*" [Cas87 (p. 1125)].

In this paper, an *oracle* U is a generator of strings such that it returns a string on each call. The generator may be deterministic or probabilistic.

An *oracle Turing machine* M is a Turing machine which has a special state called an oracle state and a special tape called an oracle tape. When M^U with an oracle U enters the oracle state, the value given by U will be written on the oracle tape in one step.

Now we define a learner as follows:

Definition 2. A *learner* $L = (K, Q)$ for a problem $D = (G, O)$ is given as follows:

1. K is a polynomial time Turing machine called the *current knowledge* of L. Given an input $x \in \Sigma^*$, it generates a sequence of operations $\sigma \in O^*$.

2. Q is an oracle Turing machine. Given the code of K, by using strings generated by the oracle, it produces codes of polynomial time Turing machines consecutively. Q runs in polynomial time with respect to the number n of symbols it has read (not necessarily the sum of the lengths of the examples).

Intuitively, when an instance is given to a learner $L = (K, Q)$, the current knowledge K tries to solve it. Hence K represents the current ability of L. On the other hand, Q generalizes a fact from examples and then improves the knowledge gradually. In this way, the learner can change itself.

Definition 3. Let $L = (K, Q)$ be a learner for a problem D and U be an oracle. The *learner at time* t *with* U is a pair $L_t = (K_t, Q)$, where K_t is the last Turing machine produced by Q^U on K until time t.

Definition 4. Let L be a learner for a problem D and P be a distribution for D. For $x \in S(D)$, let σ be the sequence of operations computed by K on x. If σ is a solution of x, we say that K *succeeds on* x. Otherwise, we say that K *fails on* x. For $0 < \epsilon < 1$, we say that L *solves* D *within error* ϵ *under* P if

$$P\left(x \in S(D) \;\middle|\; K \text{ fails on } x\right) \leq \epsilon.$$

Definition 5. Let P be a distribution of a problem D. For simplicity we denote hereafter $X = S(D)$ and $\hat{X} = \{\langle x, \sigma\rangle \in S(D) \times O^* \mid \sigma \text{ is a solution of } x\}$. A probability distribution $\hat{P}$ over $\hat{X}$ is said to be an *associated distribution* of P if it satisfies the following condition:

$$P(x) = \hat{P}(\{\langle x, \sigma\rangle \mid \sigma \text{ is a solution of } x\}) \text{ for every } x \in X.$$

Then we define $EX^{\hat{P}}$ as an oracle which generates $\beta \in \hat{X}$ according to $\hat{P}$.

Definition 6. Let L be a learner for a problem D and P be a distribution for D. We say that L *can learn* D *with* $EX^{\hat{P}}$ if there is a polynomial $p(n, m)$ satisfying the condition:

> For each $0 < \epsilon, \delta < 1$ and any $t \geq p(\frac{1}{\epsilon}, \frac{1}{\delta})$, the probability that the learner at time t with $EX^{\hat{P}}$ solves D within error ϵ under P is at least $1 - \delta$.

Let $L = (K, Q)$ be a learner for a problem D. For a sample β, we denote by $Q(\beta)$ the last Turing machine produced by Q with β. We call $Q(\beta)$ the *knowledge produced from* β. Then we define $\mathcal{K}(Q) = \{K \mid K \text{ is a partial function computed by } Q(\beta) \text{ for some sample } \beta \text{ of } D\}$. For $K' \in \mathcal{K}(Q)$ and an integer $n \geq 1$, $K'|_n$ denotes the partial function obtained by restricting its domain to the set of strings of length at most n. We define $\mathcal{K}_n(Q) = \left\{K'|_n \;\middle|\; K' \in \mathcal{K}(Q)\right\}$. We often confuse a Turing machine with the partial function computed by it.

4. Sample Size

In this section we consider how the sample size affects the learnability.

Definition 7. Let P a distribution over a problem D. For $0 < \varepsilon < 1$ we define the integer $n_{P,\varepsilon}$ as follows:

$$n_{P,\varepsilon} = \min\left\{n \in \mathcal{N} \;\middle|\; P\left(x \in X \;\middle|\; |x| \leq n\right) \geq 1 - \varepsilon\right\}$$

We denote $n_{P,\varepsilon}$ by n_ε if P is implicitly understood.

Lemma 1. $\frac{1}{-\log(1-x)} < \frac{1}{x}$ *for* $0 < x < 1$.

Lemma 2. *Let* $L = (K_0, Q)$ *be a learner for a problem* D. *Then, for any distribution* P *for* D, *any* $0 < \varepsilon < 1$ *and any integer* $l \geq 1$, *the following holds:*

$$\hat{P}^l\left(\beta \in \hat{X}^l \,\middle|\, \exists K \in \mathcal{K}(Q) \text{ such that (1) and (2) hold}\right) < |\mathcal{K}_{n_\varepsilon}(Q)| \cdot (1-\varepsilon)^l$$

(1) $P\left(x \in X \,\middle|\, K \text{ fails on } x\right) > 2\varepsilon$.

(2) If $|x_i| \leq n_\varepsilon$, *then* K *succeeds on* x_i *for* $i = 1, \cdots, l$, *where* $\beta = (\langle x_1, \sigma_1\rangle, \cdots, \langle x_l, \sigma_l\rangle)$.

Proof. We denote by $\mathcal{K}^{(1)}$ the set of all $K \in \mathcal{K}(Q)$ satisfying (1). Since

$$\begin{aligned} & P\left(x \in X \,\middle|\, |x| > n_\varepsilon \text{ and } K \text{ fails on } x\right) \\ \leq\ & P\left(x \in X \,\middle|\, |x| > n_\varepsilon\right) \\ =\ & 1 - P\left(x \in X \,\middle|\, |x| \leq n_\varepsilon\right) \\ \leq\ & 1 - (1-\varepsilon) \qquad \text{(by the definition of } n_\varepsilon) \\ =\ & \varepsilon, \end{aligned}$$

each $K \in \mathcal{K}^{(1)}$ satisfies the following condition:

(3) $P\left(x \in X \,\middle|\, |x| \leq n_\varepsilon \text{ and } K \text{ fails on } x\right) > \varepsilon$.

Hence, if we define $\mathcal{K}^{(3)}$ to be the set of $K \in \mathcal{K}(Q)$ satisfying (3), then we have

(4) $\mathcal{K}^{(1)} \subseteq \mathcal{K}^{(3)}$.

On the other hand, for any $K \in \mathcal{K}(Q)$ if we denote by $\hat{X}^l_K$ the set of $\beta \in \hat{X}^l$ satisfying (2), then

(5) $\hat{X}^l_K = \hat{X}^l_{K|_{n_\varepsilon}}$.

Therefore

$$\begin{aligned} & \hat{P}^l\left(\beta \in \hat{X}^l \,\middle|\, \exists K \in \mathcal{K}(Q) \text{ such that (1) and (2) hold }\right) \\ =\ & \hat{P}^l\left(\bigcup_{K \in \mathcal{K}^{(1)}} \hat{X}^l_K\right) \end{aligned}$$

$$
\begin{aligned}
&\leq \hat{P}^l\left(\bigcup_{K\in\mathcal{K}^{(3)}} \hat{X}_K^l\right) && (\text{ by (4) }) \\
&= \hat{P}^l\left(\bigcup_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} \hat{X}_K^l\right) && (\text{ by (5) }) \\
&\leq \sum_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} \hat{P}^l\left(\hat{X}_K^l\right) \\
&= \sum_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} \left(\hat{P}\left(\langle x,\sigma\rangle \in \hat{X} \;\middle|\; \text{if } |x|\leq n_\varepsilon \text{ then } K \text{ succeeds on } x\right)\right)^l \\
&= \sum_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} \left(P\left(x \in X \;\middle|\; \text{if } |x|\leq n_\varepsilon \text{ then } K \text{ succeeds on } x\right)\right)^l \\
&= \sum_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} \left(1-P\left(x \in X \;\middle|\; |x|\leq n_\varepsilon \text{ and } K \text{ fails on } x\right)\right)^l \\
&< \sum_{K\in\mathcal{K}_{n_\varepsilon}^{(3)}} (1-\varepsilon)^l && (\text{ by (3) }) \\
&= \left|\mathcal{K}_{n_\varepsilon}^{(3)}\right|\cdot(1-\varepsilon)^l \\
&\leq |\mathcal{K}_{n_\varepsilon}(Q)|\cdot(1-\varepsilon)^l
\end{aligned}
$$

□

Definition 8. Let $L=(K,Q)$ be a learner for a problem D. We say that Q is *consistent* if for any sample $\beta=\{\langle x_i,\sigma_i\rangle\}_{i=1}^m$, $Q(\beta)$ succeeds on $x_1,...,x_m$.

Lemma 3. *Let $L=(K_0,Q)$ be a learner for a problem D, P a distribution for D. If Q is consistent, then for any $0<\varepsilon,\delta<1$ and any integer $l\geq\frac{2}{\varepsilon}\left(\log\left|\mathcal{K}_{n_{\frac{\varepsilon}{2}}}(Q)\right|+\log\frac{1}{\delta}\right)$, the following holds:*

$$\hat{P}^l\left(\beta\in\hat{X}^l\middle|P\left(x\in X \;\middle|\; Q(\beta) \text{ fails on } x\right)>\varepsilon\right)<\delta.$$

Proof.

$$
\begin{aligned}
&\hat{P}^l\left(\beta\in\hat{X}^l\middle|P\left(x\in X \;\middle|\; Q(\beta) \text{ fails on } x\right)>\varepsilon\right) \\
&\leq \hat{P}^l\left(\beta\in\hat{X}^l \;\middle|\; \begin{array}{l}\exists K\in\mathcal{K}(Q) \text{ such that} \\ \text{(a) } P\left(x\in X \mid K \text{ fails on } x\right)>\varepsilon, \\ \text{(b) } K \text{ succeeds on } x_1,\cdots,x_l \text{ where } \beta=(\langle x_1,\sigma_1\rangle\cdots,\langle x_l,\sigma_l\rangle)\end{array}\right) \\
&< \left|\mathcal{K}_{n_{\frac{\varepsilon}{2}}}(Q)\right|\cdot\left(1-\frac{\varepsilon}{2}\right)^l && (\text{ by Lemma 2 })
\end{aligned}
$$

Since $l\geq\frac{2}{\varepsilon}\left(\log\left|\mathcal{K}_{n_{\frac{\varepsilon}{2}}}(Q)\right|+\log\frac{1}{\delta}\right)$, by Lemma 1 we can show $\left|\mathcal{K}_{n_{\frac{\varepsilon}{2}}}(Q)\right|\cdot\left(1-\frac{\varepsilon}{2}\right)^l<\delta$. □

The following theorem means that if sufficiently many examples are available then a learner achieves better.

Theorem 1. *Let $L = (K_0, Q)$ be a learner for a problem D and P be a distribution for D. Let $d(n) = \log |\mathcal{K}_n(Q)|$ for $n \geq 1$. Assume that Q is consistent. Then, for any $0 < \varepsilon, \delta < 1$, the probability that after processing a sample of size $\frac{2}{\varepsilon}\left(d(n_{max}) + \log \frac{2}{\delta}\right)$ the learner with $EX^{\hat{P}}$ solves D within error ε under P is at least $1 - \delta$, where n_{max} is the length of the longest example among the first $\frac{2}{\varepsilon}\log\frac{2}{\delta}$ examples.*

Proof. Let $m = \frac{2}{\varepsilon}\log\frac{2}{\delta}$. The probability that there is no example with length greater than $n_{\frac{\varepsilon}{2}}$ among m examples, i.e., $n_{max} < n_{\frac{\varepsilon}{2}}$, is

$$\begin{aligned} & P^m\left((x_1, \cdots, x_m) \in X^m \;\middle|\; |x_i| < n_{\frac{\varepsilon}{2}} \text{ for } i = 1, \cdots, m\right) \\ = \; & \left(P\left(x \in X \;\middle|\; |x| < n_{\frac{\varepsilon}{2}}\right)\right)^m \\ < \; & \left(1 - \frac{\varepsilon}{2}\right)^m \quad (\text{ by the definition of } n_\varepsilon \text{ }) \\ < \; & \frac{\delta}{2} \qquad\qquad (\text{by Lemma 1}) \end{aligned}$$

Hence, the probability that $\frac{2}{\varepsilon}\left(d(n_{max}) + \log\frac{2}{\delta}\right) < \frac{2}{\varepsilon}\left(d(n_{\frac{\varepsilon}{2}}) + \log\frac{2}{\delta}\right)$ holds is at most $\frac{\delta}{2}$ since $d(n)$ is nondecreasing. Therefore, by Lemma 3, the probability that after processing a sample of size $\frac{2}{\varepsilon}\left(d(n_{max}) + \log\frac{2}{\delta}\right)$ the learner solves D within error ε under P is at least $1 - \delta$. □

5. What to teach

Given examples $\langle x_1, \sigma_1\rangle$ and $\langle x_2, \sigma_2\rangle$ with $|x_1| = |x_2|$, the lengths of σ_1 and σ_2 may be extremely different. In such case, the length of the solution affects the time for processing drastically. Therefore it is not natural to discuss the learnability only by sample size in our framework. Hence Theorem 1 provides only little insight when the time for learning is of concern.

If a very long example occurs and the learner keeps processing the example, it may just exhaust time without getting any further information. In the case that the learner is not required to solve the problem perfectly, it seems to be reasonable to ignore long examples. However, if the length of the examples which can be ignored is not known, the learner may get in trouble. Therefore it is important for

the learner to be taught the length of the examples to process. Theorem 2 in this section is based on this observation.

Let $L = (K, Q)$ be a learner. For an integer $n \geq 1$, $Q^{[n]}$ denotes an algorithm that runs in the following way: Given an example α of length m, it simulates Q on α while Q is reading the first n symbols of α, but if $m > n$ then it ignores the rest of α without reading and continues on the next example. Hence at most n symbols of α can be read.

For functions $v(n)$ and $g(n)$, a *crescendo algorithm* $Q[v, g]$ is an algorithm described as follows:

begin
  $i \leftarrow 1$;
  repeat
    Run $Q^{[v(i)]}$ until $g(i)$ input symbols are processed;
    $i \leftarrow i + 1$;
  forever
end

Crescendo Algorithm

Intuitively speaking, $Q[v, g]$ tries to cope with examples by increasing the difficulty.

Definition 9. We say that an associated distribution $\hat{P}$ of P is *polynomially bounded* if there is a polynomial $s(n)$ such that $|\langle x, \sigma \rangle| \leq s(|x|)$ for all $\langle x, \sigma \rangle \in \hat{X}$ with $\hat{P}(\langle x, \sigma \rangle) \neq 0$.

The reason why we put the polynomially bounded assumption on $\hat{P}$ is that too long solutions compared with their instances are not appropriate since K of a learner $L = (K, Q)$ is assumed to generate solutions in polynomial time. It is also not natural to consider only optimal solutions as in [Nat89].

We also consider the following restriction on distributions.

Definition 10. Let P be a distribution for a problem D. We say that P is *polynomially decreasing* if there is a polynomial $f(n)$ such that $n_\varepsilon \leq f(\frac{1}{\varepsilon})$ for $0 < \varepsilon < 1$. This means that for all ε the probability that an instance of length greater than $f(\frac{1}{\varepsilon})$ occurs is at most ε.

Remark 1. There is a distribution which is not polynomially decreasing. For example, consider the distribution $P(x)$ on $\{0\}^*$ defined by

$$P(x) = \begin{cases} \frac{6}{\pi^2}\frac{1}{k^2} & \text{if } |x| = 2^k \qquad (k = 1, 2, \cdots) \\ 0 & \text{otherwise} \end{cases}$$

Theorem 2. *Let $L = (K, Q)$ be a learner for a problem D and P be a distribution for D and $\hat{P}$ be a polynomially bounded associated distribution of P. Assume that the following conditions are satisfied:*

(1) Q is consistent.

(2) There is a polynomial $d(n)$ such that $|\mathcal{K}_n(Q)| \leq 2^{d(n)}$ for $n \geq 1$.

(3) P is polynomially decreasing.

Then $\tilde{L} = (K, Q[v, g])$ with $EX^{\hat{P}}$ can learn D under P for some polynomials $v(n)$ and $g(n)$.

Remark 2. Even in the case that $\Sigma = \{0, 1\}$ and the output of K are only 0 or 1, $|K_n(Q)|$ may be as large as double exponential in n. Hence, the second condition means that Q has a kind of ability to generalize facts from examples.

Proof of Theorem 2. Since $\hat{P}$ is polynomially bounded, there is a polynomial $s(n)$ satisfying Definition 9. Then the following claims hold:

Claim 1. For any $0 < \varepsilon,\ \delta < 1$ and any integer $l \geq \frac{2}{\varepsilon}\left(\log\left|\mathcal{K}_{n_{\varepsilon/2}}(Q)\right| + \log\frac{1}{\delta}\right)$, the following holds:

$$\hat{P}^l\left(\beta \in \hat{X}^l \,\middle|\, P\left(x \in X \;\middle|\; Q^{[s(n_{\varepsilon/2})]}(\beta) \text{ fails on } x\right) > \varepsilon\right) < \delta$$

Proof.

$$\hat{P}^l\left(\beta \in \hat{X}^l \,\middle|\, P\left(x \in X \;\middle|\; Q^{[s(n_{\varepsilon/2})]}(\beta) \text{ fails on } x\right) > \varepsilon\right)$$

$$\leq \hat{P}^l\left(\beta \in \hat{X}^l \;\middle|\; \begin{array}{l} \exists K \in \mathcal{K}(Q) \text{ such that} \\ \text{(a) } P\left(x \in X \mid K \text{ fails on } x\right) > \varepsilon, \\ \text{(b) if } |\langle x_i, \sigma_i, \rangle| \leq s(n_{\frac{\varepsilon}{2}}) \text{ then} \\ K \text{ succeeds on } x_i \text{ for } i = 1, \cdots l, \text{ where } \beta = (\langle x_1, \sigma_1\rangle, \cdots, \langle x_l, \sigma_l\rangle) \end{array}\right)$$

$$\leq \hat{P}^l \left(\beta \in \hat{X}^l \;\middle|\; \begin{array}{l} \exists K \in \mathcal{K}(Q) \text{ such that} \\ \text{(a) } P\left(x \in X \mid K \text{ fails on } x\right) > \varepsilon, \\ \text{(b) if } |x_i| \leq n_{\frac{\varepsilon}{2}} \text{ then} \\ K \text{ succeeds on } x_i \text{ for } i = 1, \cdots l, \text{ where } \beta = (\langle x_1, \sigma_1 \rangle, \cdots, \langle x_l, \sigma_l \rangle) \end{array} \right)$$

$$< \left| \mathcal{K}_{n_{\frac{\varepsilon}{2}}}(Q) \right| \cdot \left(1 - \frac{\varepsilon}{2}\right)^l \quad < \quad \delta.$$

Claim 2. There is a polynomial $p(n, m)$ such that for every $0 < \varepsilon, \delta < 1$, at any time $t \geq p(\frac{1}{\varepsilon}, \frac{1}{\delta})$ the learner $L[\varepsilon] = (K, Q^{[s(n_{\varepsilon/2})]})$ with the oracle $EX^{\hat{P}}$ solves D within error ε under P with probability at least $1 - \delta$.

Proof. By Claim 1, the probability that the learner $L[\varepsilon]$ which has processed a sample of size $l(\varepsilon, \delta) = \frac{2}{\varepsilon}\left(d\left(n_{\frac{\varepsilon}{2}}\right) + \log\frac{1}{\delta}\right)$ solves D within error ε under P is at least $1 - \delta$. It suffices to estimate the running time that $Q^{[s(n_{\varepsilon/2})]}$ takes until it finishes processing a sample of size $l(\varepsilon, \delta)$. Since $Q^{[s(n_{\varepsilon/2})]}$ can read at most $s(n_{\frac{\varepsilon}{2}}) \cdot \frac{2}{\varepsilon}\left(d\left(n_{\frac{\varepsilon}{2}}\right) + \log\frac{1}{\delta}\right)$ symbols, it follows from (2) and (3) that the running time is bounded by some polynomial $p(\frac{1}{\varepsilon}, \frac{1}{\delta})$.

Without loss of generality, we may assume that $s(n)$ and $f(n)$ are nondecreasing. Let $v(n) = s(f(2n))$, $l(n) = 2n\left(d(f(2n)) + \log n\right)$ and $g(n) = v(n) \cdot l(n)$. Then since in the kth phase $Q^{[v(k)]}$ reads $g(k)$ symbols, it processes at least $l(k)$ examples. By the proof of Claim 2, the learner $Q[v, g]$ after the kth phase solves D within error $\frac{1}{k}$ under P with probability at least $1 - \frac{1}{k}$.

Until the end of the kth phase, $Q[v, g]$ reads $\sum_{j=1}^{k} g(j)$ symbols. Assume that Q runs in time $c(n)$ for some polynomial $c(n)$. Then it takes $r(k) = c(\sum_{j=1}^{k} g(j))$ time. Obviously, $r(k)$ is a polynomial in k. For each $0 < \varepsilon, \delta < 1$, we choose the integer k such that $\frac{1}{k} \leq \varepsilon\delta < \frac{1}{k-1}$. Since $\frac{1}{k} < \varepsilon$, δ and $r(k) \leq r(\frac{1}{\varepsilon\delta} + 1)$, the theorem holds. □

Remark 3. Claim 2 asserts that for each ε if we could *teach* the maximum length of instances to be processed then the learner will attain the given error bound in polynomial time. Such probability is also polynomially related to the time.

6. Teaching with a Manual

The purpose of this section is to introduce the following notion.

Definition 11. Let L be a learner for a problem D and P be a distribution for D. We say that D *can be taught to* L *under* P if there are a polynomial $p(m)$ and an oracle T which generates examples of D such that for each $0 < \varepsilon < 1$ and $t \geq p(\frac{1}{\varepsilon})$ the learner at time t with T solves D within error ε under P. We call the oracle T a *teaching manual* for D.

We now give an example of teaching with a manual.

Let Σ be an alphabet satisfying $\Sigma \cap \{0,1\} = \emptyset$ and let $\Sigma' = \Sigma \cup \{0,1\}$. Then we define the function $o_y : \Sigma'^* \to \Sigma'^*$ (resp. o_n) by $o_y(x) = x1$ (resp. $o_n(x) = x0$). For a language $A \subseteq \Sigma^*$ we consider the problem $D_A = (G_A, O)$ called the *language identification problem for* A, where $O = \{o_y, o_n\}$ and the goal function G_A is defined by

$$G_A(y) = \begin{cases} 1 & \text{if } y = xi \text{ for some } x \in \Sigma^* \text{ and } i \in \{0,1\} \\ & \text{and } x \in A \Leftrightarrow i = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let G be an unambiguous context-free grammar. We denote the set of sentential forms of G by $U = \{\alpha \mid S \stackrel{*}{\Rightarrow} \alpha\}$, where S is the start symbol of G. For $\alpha \in U$, we denote $L(\alpha) = \{x \in \Sigma^* \mid \alpha \stackrel{*}{\Rightarrow} x\}$, and let $D_{(\alpha)}$ be the language identification problem for $L(\alpha)$, where Σ is the set of terminal symbols of G. Obviously, $(U, \stackrel{*}{\Rightarrow})$ is a partially ordered set. It is also easy to see that for sentential forms $\alpha, \beta \in U$ there is a sentential form γ satisfying (1) and (2).

(1) $\gamma \stackrel{*}{\Rightarrow} \alpha$, $\gamma \stackrel{*}{\Rightarrow} \beta$.

(2) If $\gamma' \stackrel{*}{\Rightarrow} \alpha$, $\gamma' \stackrel{*}{\Rightarrow} \beta$, then $\gamma' \stackrel{*}{\Rightarrow} \gamma$.

Moreover, such γ is unique since the grammar G is unambiguous. Therefore we denote such γ by $\alpha \vee \beta$.

The following lemma asserts that the least upper bound $\alpha \vee \beta$ is computable in polynomial time.

Lemma 4. *For any $\alpha, \beta \in U$, $\alpha \vee \beta$ can be computed in polynomial time with respect to $|\alpha|$, $|\beta|$.*

Proof. First construct the parse trees for α and β. Then march up these parse trees simultaneously to pick off points common to both of them. □

Let β be a sentential form in U. We define a learner $L_\beta = (K_\beta, Q_G)$ as follows: First let K_β be the following algorithm.

K_β : **input** x;
if $\beta \stackrel{*}{\Rightarrow} x$ **then output** o_y; **else output** o_n;

Then we define the algorithm Q_G as follows:

Q_G : **input** K_β;
repeat
input($\langle x, \sigma \rangle$);
if $\sigma = o_y$ **then**
$\beta := \beta \vee x$;
output K_β;
forever

Theorem 3. *Let α be a sentential form in U and P be any distribution for $D_{(\alpha)}$. Then $D_{(\alpha)}$ can be taught to $L_\beta = (K_\beta, Q_G)$ under P if $\alpha \stackrel{*}{\Rightarrow} \beta$.*

Proof. It is not hard to show that there are two words $v, w \in \Sigma^*$ such that $L(v \vee w) = L(\alpha)$. Then we can show that $L((\beta \vee v) \vee w) = L(\alpha)$ since $\alpha \stackrel{*}{\Rightarrow} \beta$. Therefore after processing $\langle v, o_y \rangle$ and $\langle w, o_y \rangle$, Q_G produces K with no error. □

Definition 12. For a problem D, a *naïve learner* $L = (K, Q)$ for D is defined as follows: K has a list of examples. Given an instance x, K searches the list for the example corresponding to x. If the example $\langle x, \sigma \rangle$ is found, K outputs σ. Otherwise, K outputs the null string. Q keeps all examples obtained so far together with the examples in K. At each time when a new example comes, Q constructs a list of examples and output K' with this list.

Proposition 1. *Let $L = (K_0, Q)$ be a naïve learner for a problem D, where the list in K_0 is empty. Let P be a distribution for D. If D can be taught to L under P, then P is polynomially decreasing.*

Proof. By the assumption, there are a polynomial $p(m)$ and a teaching manual T such that for each $0 < \varepsilon < 1$ and $t \geq p(\frac{1}{\varepsilon})$ the learner at time t with T solves D within error ε under P. Now we assume that P is not polynomially decreasing. Then by the definition, for the polynomial $p(m)$ we can take ε with $n_{P,\varepsilon} > p(\frac{1}{\varepsilon})$. Since the current knowledge K_0 is empty and since L is a naïve learner, it must read an example with an instance whose length is not less than $n_{P,\varepsilon}$ in order to solve D within error ε. But it is impossible by the time $t = p(\frac{1}{\varepsilon})$. □

Definition 13. Let P be a distribution for a problem D. We say that P is *sparse* if there is a polynomial $p(n)$ such that $|\{x \in S(D) \mid P(x) > 0,\ |x| \leq n, \}| \leq p(n)$ for each $n > 0$.

Theorem 4. *Let $L = (K, Q)$ be a naïve learner for a problem D and P be a distribution for D. We assume that each instance $x \in S(D)$ with $P(x) > 0$ has a solution whose length is polynomially related to $|x|$. If P is sparse and polynomially decreasing, D can be taught to L under P.*

Proof. We consider the following teaching manual: It generates pairs $\langle x, \sigma \rangle$ for all instances $x \in S(D)$ with $P(x) > 0$ in the increasing order of $|x|$, where σ is a solution of x whose length is polynomially related to $|x|$. It should be noticed that if all examples for $V_\varepsilon = \{x \in S(D) \mid P(x) > 0, |x| < n_\varepsilon\}$ are fed into Q then the naïve learner L solves D within error ε under P.

Now we estimate the time t required to process all examples for V_ε. Since P is sparse, there is a polynomial $q(n)$ satisfying $|V_\varepsilon| \leq q(n_\varepsilon)$. Then observe that the total length of examples for V_ε is bounded by some polynomial with respect to n_ε. since the length of example $\langle x, \sigma \rangle$ is polynomially related to $|x|$. Finally, since P is polynomially decreasing, n_ε is also bounded by some polynomial with respect to $\frac{1}{\varepsilon}$. Thus the theorem holds. □

7. Conclusion

We made a framework for learning that allows arguments from teaching. The results in this paper show that this framework is suitable for capturing our intuition about learning mechanism.

References

[Ang88] D. Angluin, Queries and concept learning, *Machine Learning*, **2** (4), 319–342, 1988.

[Lai89] P. Laird, A survey of computational learning theory, Technical Report RIA-89-01-07-0, Artificial Intelligence Research Branch, NASA Ames Research Center, January 1989.

[Nat89] B.K. Natarajan, On learning from exercises, *Proceedings of the 2nd Annual Workshop on Computational Learning Theory*, 72–87, 1989.

[NT88] B.K. Natarajan and P. Tadepalli, Two new frameworks for learning, *Proceedings of the 5th International Workshop on Machine Learning*, 402–415, 1988.

[Pit89] L. Pitt, Inductive inference, DFAs, and computational complexity, Technical Report UIUCDCS-R-89-1530, Department of Computer Science, University of Illinois at Urbana-Champaign, July 1989.

[Cas87] J. Case, *Encyclopedia of Artificial Intelligence*, S.C. Shapiro (Ed.), Wiley-Interscience, 1987.

[Val84] L.G. Valiant, A theory of the learnable, *Communications of the ACM*, **27** (11), 1134–1142, 1984.